On the Capacity of Computation Broadcast

Hua Sun and Syed A. Jafar

Abstract

The two-user computation broadcast problem is introduced as the setting where User 1 wants message W_1 and has side-information W_1', User 2 wants message W_2 and has side-information W_2', and (W_1, W_1', W_2, W_2') may have arbitrary dependencies. The rate of a computation broadcast scheme is defined as the ratio $H(W_1, W_2)/H(S)$, where S is the information broadcast to both users to simultaneously satisfy their demands. The supremum of achievable rates is called the capacity of computation broadcast C_{CB}. It is shown that $C_{CB} \leq H(W_1, W_2)/\left[H(W_1|W_1') + H(W_2|W_2') - \min\left(I(W_1; W_2, W_2'|W_1'), I(W_2; W_1, W_1'|W_2')\right)\right]$. For the linear computation broadcast problem, where W_1, W_1', W_2, W_2' are comprised of arbitrary linear combinations of a basis set of independent symbols, the bound is shown to be tight. For non-linear computation broadcast, it is shown that this bound is not tight in general. Examples are provided to prove that different instances of computation broadcast that have the same entropic structure, i.e., the same entropy for all subsets of $\{W_1, W_1', W_2, W_2'\}$, can have different capacities. Thus, extra-entropic structure matters even for two-user computation broadcast. The significance of extra-entropic structure is further explored through a class of non-linear computation broadcast problems where the extremal values of capacity are shown to correspond to minimally and maximally structured problems within that class.

Hua Sun (email: hua.sun@unt.edu) is with the Department of Electrical Engineering at the University of North Texas. Syed A. Jafar (email: syed@uci.edu) is with the Center of Pervasive Communications and Computing (CPCC) in the Department of Electrical Engineering and Computer Science (EECS) at the University of California Irvine. This work was presented in part at ICC 2019.

1 Introduction

In the modern era of data science, machine learning and internet of things, communication networks are increasingly used for distributed computing applications, where multiple parties process and exchange information for various computational tasks [1–8]. The changing paradigm brings with it new challenges in network information theory. Distinctive aspects of these computational communication networks include strong dependencies between information flows and an abundance of side-information. With a few notable exceptions such as [9–17], the communication network models most commonly studied in information theory, in various elemental forms ranging from multiple access and broadcast to relay and interference networks, with and without side-information, tend to focus on settings with *independent* messages. Yet, the shared mission across network nodes in distributed computing applications necessarily creates significant dependencies, not only among message flows, but also in the side-information available to each node based on its history of prior computations. Within these dependencies lies the potential for further innovations in communication and computing. A fundamental understanding of this potential requires the machinery of network information theory, but with a renewed focus on information dependencies and side-information. As a step in this direction, in this work we introduce the problem of computation broadcast.

While in this work we restrict our attention to $K = 2$ users, in general we envision the computation broadcast (CB) problem as comprised of K users (receivers) who desire messages $W_1, W_2, \cdots, W_K$, and have prior side-information $W_1', W_2', \cdots, W_K'$, respectively. A centralized transmitter with full knowledge of $(W_k, W_k', k \in [K])$ broadcasts the same information S to all receivers in order to simultaneously deliver their desired messages. The salient feature of computation broadcast is the dependence among $(W_k, W_k', k \in [K])$ modeled by their joint distribution, which may be arbitrary.

The rate of computation broadcast is defined as, $R = H(W_1, \cdots, W_K)/H(S)$, i.e., ratio of the total number of bits of all desired messages to the number of bits of broadcast information S that satisfies all demands. The supremum of achievable rates is called the capacity of computation broadcast, C_{CB}. The goal is to characterize C_{CB}.

The computation broadcast problem may be seen as a generalization of the index coding problem [18, 19] that allows arbitrary dependencies among desired messages and side-informations. Prior works in this direction include [20, 21]. Reference [20] restricts the messages to be independent and requires each side-information to be a linear combination of message symbols, which is a special case of computation broadcast. The problem formulation of [21] allows the messages to be arbitrarily correlated while the side-informations are comprised of message symbols, which is another special case of computation broadcast. Also, when we have $K = 2$ users and $W_1' = W_2$ and $W_2' = W_1$, the computation broadcast problem reduces to the classic butterfly network problem with possibly correlated sources [22, 23].

The dependence between desired messages and side-informations imparts a unique structural aspect to the computation broadcast problem that makes it highly non-trivial. Structure has long been recognized as both the boon and bane of network information theory [24–28]. When optimally exploited, structure can have tremendous benefits in multiterminal settings, a fact underscored by recurring observations ranging from Korner and Marton's computation work in [24] to the recent burst of activity in interference alignment [29,30]. On the other hand, the random coding arguments that are the staple of classical information theory, tend to fall short when structural concerns take center stage, and less tractable combinatorial alternatives are required. Structure itself is a

nebulous concept that has thus far defied a precise understanding. Somewhat surprisingly, these deeper themes resurface even in the basic 2 user setting explored in this work. On the downside this potentially makes even the 2 user computation broadcast problem intractable in general. On the upside, the 2 user computation broadcast presents one of the simplest arenas to face these challenges that are of tremendous theoretical and practical significance.

Our contributions in this paper are summarized as follows. We start with a general converse bound for the capacity of 2 user computation broadcast,

$$C_{CB} \leq H(W_1, W_2) / \left[H(W_1 \mid W_1') + H(W_2 \mid W_2') - \min \left(I(W_1; W_2, W_2' \mid W_1'), I(W_2; W_1, W_1' \mid W_2') \right) \right].$$

When the dependency is linear, i.e., when W_1, W_1', W_2, W_2' are comprised of arbitrary linear combinations of a basis set of independent symbols, then this bound is shown to be tight. However, in general the bound is not tight, and the structure of the dependence between W_1, W_1', W_2, W_2', becomes important. Recall that the dependence is completely described by the joint distribution of (W_1, W_1', W_2, W_2') which can be arbitrary. Some of this structure can be captured through entropic constraints, i.e., the joint entropies of all subsets of (W_1, W_2, W_1', W_2'). One might optimistically expect that only this entropic structure would be essential to the problem, and furthermore that Shannon information inequalities might suffice to characterize the optimal $H(S)$. However, as it turns out on both counts the optimism is invalidated. Specifically, we show two instances of computation broadcast that have the same entropic description, yet different capacity characterizations. Evidently, extra-entropic structure matters even for 2-user computation broadcast. In order to further understand the significance of such extra-entropic structure, we explore a class of computation broadcast problems called 'matching' problems where, conditioned on each realization of the independent side-informations W_1', W_2', there is a perfect matching between the possible realizations of desired messages W_1, W_2. For this class of problems we identify upper and lower bounds on capacity. The bounds provide insights into certain types of extremal structures that are either beneficial or detrimental to capacity. The beneficial extremes are found to be maximally structured and for these settings the capacity upper bound is shown to be tight. Conversely, the detrimental extremes are found to be minimally structured and for these settings the capacity lower bound is shown to be tight. Remarkably, linear dependencies are maximally structured, while random coding solutions are asymptotically optimal for minimally structured settings in the limit of large alphabet sizes.

Notation: For a positive integer m, we use the notation $[m] = \{1, 2, \cdots, m\}$. Bold symbols are used to represent matrices.

2 Problem Statement and Preliminaries

Define random variables $(w_1, w_1', w_2, w_2') \in \mathcal{W}_1 \times \mathcal{W}_1' \times \mathcal{W}_2 \times \mathcal{W}_2'$, drawn according to an arbitrary joint distribution $P_{w_1, w_1', w_2, w_2'}$. All 4 alphabet sets are discrete with finite cardinality bounded by $2^{\ell_{\max}} < \infty$, i.e., it takes no more than a finite number ($\ell_{\max}$) of bits to perfectly specify any w_i, w_i', $i \in \{1, 2\}$.

2.1 Complete (Structural) Formulation

The complete formulation of the computation broadcast problem is presented as follows.

$$
\begin{aligned}
R_L^* \triangleq \sup_{P_{S|W_1,W_2,W_1',W_2'}} & \frac{H(W_1,W_2)}{H(S)} \\
\text{such that} \quad & H(W_1 \mid W_1', S) = 0 && (1)\\
& H(W_2 \mid W_2', S) = 0 && (2)\\
& [(W_1(l), W_1'(l), W_2(l), W_2'(l))]_{l=1}^{L} \overset{\text{i.i.d.}}{\sim} P_{w_1,w_1',w_2,w_2'} && (3)
\end{aligned}
$$

As indicated in (3), W_1, W_2, W_1', W_2' denote L length extensions of w_1, w_1', w_2, w_2', i.e., W_1, W_1', W_2, W_2' are sequences of length L, such that the sequence of tuples $[(W_1(l), W_1'(l), W_2(l), W_2'(l))]_{l=1}^{L}$ is produced i.i.d. according to $P_{w_1,w_1',w_2,w_2'}$. Because the structure of the problem is completely captured in (3), we refer to this problem formulation as the complete, or structural formulation. L is called the block length. $H(S)$ is the expected amount of broadcast information. Condition (1) is the decoding constraint of User 1, i.e., after receiving the broadcast information S, User 1 is able to decode his desired message W_1 with the help of the side-information W_1', with zero probability of error. Similarly, condition (2) is the decoding constraint of User 2. Note that $H(W_1, W_2)$ is already specified by the problem statement, so maximizing R_L^* is the same as minimizing the broadcast cost, $H(S)$. The ratio $H(W_1, W_2)/H(S)$ for a computation broadcast scheme is called its achievable rate. R_L^* is the supremum of achievable rates for a given block length L. The supremum of R_L^* across all $L \in \mathbb{N}$, is called the capacity of computation broadcast,

$$C_{CB} \triangleq \sup_{L \in \mathbb{N}} R_L^*. \tag{4}$$

2.2 Relaxed (Entropic) Formulation

Recall that the structure of the dependence between message and side-information random variables is defined by Condition (3). Some of this structure can be captured in terms of the entropies of all subsets of $\{w_1, w_2, w_1', w_2'\}$. Limited to just these entropic constraints we obtain the following relaxed problem formulation.

$$
\begin{aligned}
\overline{R}_L^* \triangleq \sup_{\bar{P}_{W_1,W_2,W_1',W_2',S}} & \frac{H(W_1,W_2)}{H(S)} \\
\text{such that} \quad & H(W_1 \mid W_1', S) = 0 && (5)\\
& H(W_2 \mid W_2', S) = 0 && (6)\\
& H(W_*) = LH(w_*), \qquad \forall W_* \subset \{W_1, W_2, W_1', W_2'\} && (7)\\
& w_1, w_2, w_1', w_2' \sim P_{w_1,w_2,w_1',w_2'} && (8)
\end{aligned}
$$

where w_* is obtained by replacing upper case W with lower case w in W_*. For example, if $W_* = (W_1, W_2')$, then $w_* = (w_1, w_2')$. Note that (W_1, W_2, W_1', W_2') are arbitrary random variables that only need to satisfy the same entropic constraints as the L-length extensions of (w_1, w_2, w_1', w_2'), according to (7). In particular, it is no longer necessary for (W_1, W_2, W_1', W_2') to have the same distribution as (w_1, w_2, w_1', w_2'), even for $L = 1$. Furthermore, since the entropic region is a cone [31],

we must have $\overline{R}_L^* = \overline{R}_1^*$, where $\overline{R}_1^*$ is the value of $\overline{R}_L^*$ for $L = 1$. Since L is a trivial scaling factor, let us fix $L = 1$, and define

$$\overline{C}_{CB} \triangleq \sup_{L \in \mathbb{N}} \overline{R}_L^* = \overline{R}_1^* \tag{9}$$

$\overline{C}_{CB}$ is of interest mainly for two reasons. First, because it serves as a bound for C_{CB}, i.e.,

$$C_{CB} \leq \overline{C}_{CB}. \tag{10}$$

This is true because all the entropic constraints (7) are implied by Condition (3), so we must have $R_L^* \leq \overline{R}_L^*$ which in turn implies that $C_{CB} \leq \overline{C}_{CB}$. The second reason is that the tightness of the bound (10) reveals the extent to which capacity is determined by structural constraints that are not captured by the entropic formulation. This extra-entropic structure may be a topic of interest by itself.

2.3 Equivalence of zero-error and ϵ-error

While we consider the zero-error capacity formulation, it turns out that for the computation broadcast problem, it is not difficult to prove that zero-error capacity is the same as ϵ-error capacity, as stated in the following theorem. For this theorem we use the specialized notation C_{CB}^0 to denote zero-error capacity, and C_{CB}^ϵ to denote ϵ-error capacity.

Theorem 1 *For the computation broadcast problem, zero error capacity, C_{CB}^0, is equal to ϵ-error capacity, C_{CB}^ϵ .*

Proof: Since the ϵ-error capacity is C_{CB}^ϵ, for any arbitrarily small $\delta > 0$, there must exist an ϵ-error scheme that achieves rate $R_\epsilon = C_{CB}^\epsilon - \delta$, so that broadcasting $LH(w_1, w_2)/R_\epsilon$ bits is sufficient to satisfy both users' demands with probability at least $1 - \epsilon$, and $\epsilon \to 0$ as $L \to \infty$. Since the encoder knows all messages, side-informations and decoding functions, it also knows when either decoding function will produce an erroneous output. In those cases, the encoder can simply use uncoded broadcast to send both messages using no more than $2L\ell_{\max}$ bits. One extra bit, say the first bit, is used to indicate when uncoded transmission takes place. Thus we have a zero-error scheme, and the rate achieved is

$$\frac{LH(w_1, w_2)}{(1-\epsilon)(LH(w_1, w_2)/R_\epsilon) + \epsilon(2L\ell_{\max}) + 1} \overset{L\to\infty}{\longrightarrow} R_\epsilon \tag{11}$$

Since the rate $R_\epsilon = C_{CB}^\epsilon - \delta$ is asymptotically achievable with zero probability of error for any $\delta > 0$, the zero error capacity C_{CB}^0, which is the supremum of rates achievable with zero-error, cannot be less than C_{CB}^ϵ. At the same time, C_{CB}^0 cannot be more than C_{CB}^ϵ because allowing ϵ decoding error cannot hurt. Therefore, we must have $C_{CB}^0 = C_{CB}^\epsilon$. ■

2.4 Introductory Examples

2.4.1 Example 1: The Butterfly Network

For our first example, consider $(w_1, w_2, w_1', w_2') = (A, B, B, A)$, where A, B are i.i.d. uniform over some finite field $\mathbb{F}_q$. This is the butterfly network that is one of the most recognizable settings for network coding and index coding. The solution is also well known. The capacity is 2 and is achieved

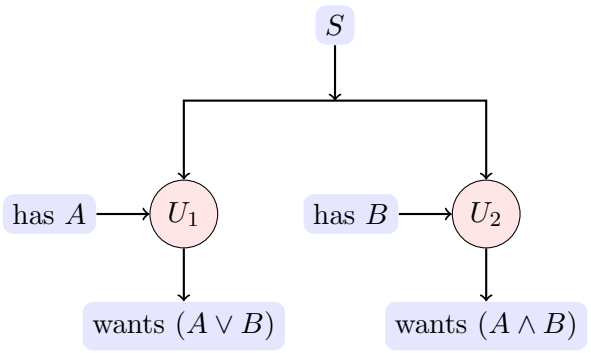
by broadcasting $S = A + B$ (the addition is in $\mathbb{F}_q$) to simultaneously satisfy both users' demands. The example can be generalized to (w_1, w_2, w_1', w_2') where w_1 is a function of w_2' and w_2 is a function of w_1'. In this case, we need a codeword of $H(w_1 \mid w_1')$ bits to satisfy User 1, corresponding to the bin index when w_1 is binned according to Slepian-Wolf coding (does not need the knowledge of w_1' at the encoder). These bits are known to User 2, because User 2 knows the binning function as well as w_2', and w_1 is a function of w_2'. Similarly, we need $H(w_2 \mid w_2')$ bits to satisfy User 2, and these bits are known to User 1. Therefore, we can choose S as the bitwise XOR of the two codewords (padding with additional zeros if needed so we have equal number of bits for both codes), which satisfies both users' demands. So the capacity for this case is $C_{CB} = \frac{H(w_1,w_2)}{\max(H(w_1|w_1'),H(w_2|w_2'))}$.

2.4.2 Example 2: A Minimal Linear Dependence Setting

Consider w_1, w_2, w_1', w_2', all in $\mathbb{F}_q$, with a 'minimal' dependence among them in the sense that any three of these four random variables are independent and uniform, while the dependence arises due to the constraint $w_1 + w_2 + w_1' + w_2' = 0$. In this case, the capacity is still 2, and it is achieved by broadcasting $S = w_1 + w_1'$, which simultaneously satisfies both users. This example is inspired by a general capacity achieving scheme for linear computation broadcast problems that is developed in this work.

2.4.3 Example 3: A Binary AND/OR Problem

For our third example, let us consider a non-linear computation broadcast problem, where we have $(w_1, w_2, w_1', w_2') = (A \vee B, A \wedge B, A, B)$, and A, B are independent uniform binary random variables. The notations $\vee, \wedge$ represent the logical OR and AND operations, respectively. Thus, User 1 knows A and wants $A \vee B$, while User 2 knows B and wants $A \wedge B$.



(S_{AB})	$B=0$	$B=1$
$A=0$	S_{00}	S_{01}
$A=1$	S_{10}	S_{11}

and $S_{00} \neq S_{01}$, $S_{01} \neq S_{11}$

$(\text{Optimal } S_{AB})$	$B=0$	$B=1$
$A=0$	0	1
$A=1$	0	0

Figure 1: Toy example where User 1 has side-information A and wants to receive $(A \vee B)$ while User 2 has side-information B and wants $(A \wedge B)$. The optimal solution broadcasts only 0.5 bits/symbol to simultaneously satisfy both users' demands.

Note that the desired message and available side-information are not independent. Also note that in order to satisfy User 1 alone, we need at least $H(A \vee B|A) = 0.5$ bits/symbol. Similarly, in order to satisfy User 2 alone, we need at least $HA \wedge B|B) = 0.5$ bits/symbol. But what is the most efficient way to satisfy both users' demands simultaneously? Surprisingly, 0.5 bits/symbol is also sufficient to simultaneously satisfy the demands of both users. This is accomplished as follows. Let us first consider block length $L = 1$ and let S_{AB} represent the value of the broadcast symbol S as a function of the values of A and B. Now, when $A = 0$, then $B = 0$ and $B = 1$ produce different

values of $A \vee B$. In order for User 1 to be able to distinguish between the two possibilities, we must have $S_{00} \neq S_{01}$. Similarly, when $B = 1$, then $A = 0$ and $A = 1$ produce different values of $A \wedge B$, so that in order to satisfy User 2's demand, we must have $S_{01} \neq S_{11}$. Subject to these two constraints, i.e., $S_{00} \neq S_{01}$ and $S_{01} \neq S_{11}$ let us assign values to S_{AB} to minimize the number of bits needed to send S_{AB} to both users using Slepian-Wolf coding, i.e., $\max(H(S_{AB}|A), H(S_{AB}|B))$. The solution for this toy problem gives us $S_{AB} = 1$ if $(A, B) = (0, 1)$ and $S_{AB} = 0$ otherwise. Note that

$$H(S_{AB}|A) \quad = \quad P(A=0)H(S_{AB}|A=0) + P(A=1)H(S_{AB}|A=1) = 0.5 \text{ bits/symbol} \quad (12)$$

and similarly, $H(S_{AB}|B) = 0.5$ bits/symbol. Remarkably, Slepian-Wolf coding allows us to satisfy both users' demands by sending only 0.5 bits/symbol. Specifically, we consider larger blocks of length $L \to \infty$, randomly bin the 2^L realizations of S_{AB}^L into $2^{L(0.5+\epsilon)}$ bins, and broadcast only[1] the bin index as S which requires $H(S) \leq L(0.5+\epsilon)$ bits. Because of the joint asymptotic equipartition property (AEP), User 1 finds a unique S_{AB}^L sequence that is jointly typical with its side-information sequence A^L with high probability, while User 2 finds a unique S_{AB}^L sequence that is jointly typical with its side-information sequence B^L with high probability. Thus, rates arbitrarily close to 0.5 bits per source symbol are achievable.[2] Remarkably, 0.5 bits per source symbol is also optimal because

$$\begin{aligned} H(A \vee B \mid A) &= P(A=0)H(A \vee B \mid A=0) + P(A=1)H(A \vee B \mid A=1) && (13) \\ &= 0.5H(B) + 0.5(0) = 0.5 \text{ bits/symbol} && (14) \end{aligned}$$

and similarly $H(A \wedge B \mid B) = 0.5$ bits /symbol. Thus, at least 0.5 bits/symbol is needed to satisfy either user alone. Fig. 1 illustrates this toy example.

2.4.4 Example 4: A Ternary AND/OR Problem

In order to emphasize the difficulty of the computation broadcast problem in general, suppose we only slightly modify the example as follows. Suppose now that $A, B \in \{0, 1, 2\}$ are i.i.d. uniform 3-ary random variables. As the natural extension of the previous example to 3-ary symbols, let us now define $A \vee B$ as 0 if $(A, B) = (0, 0)$ and 1 otherwise. Similarly, define $A \wedge B$ as 1 if $(A, B) = (1, 1)$ and 0 otherwise. As before, User 1 knows A and wants $A \vee B$ while User 2 knows B and wants $A \wedge B$. Even though this problem is only slightly modified from the previous example for which the capacity was characterized, the capacity for this modified case seems to be a challenging open problem.

2.5 Two Classes of Computation Broadcast Problems

There are two main classes of computation broadcast problems that we explore in this work – linear settings and matching problems. These classes are defined next.

[1]Note that directly setting $S = S_{AB}$ and operating over block length $L = 1$ is the best solution for $L = 1$, i.e., $R_1^* = H(w_1, w_2)/H(S_{AB}) = H(w_1, w_2)/(2 - \frac{3}{4}\log_2(3))$. However, this is not capacity-achieving because $C_{CB} = H(w_1, w_2)/2 > R_1^*$. The example shows explicitly why the problem formulation in (1)-(2) in multi-letter form (arbitrarily large $L \in \mathbb{N}$) cannot be trivially single-letterized by restricting to the case $L = 1$.

[2]Slepian-Wolf coding with distributed side-information in general may need ϵ-error. However, in our case, since the encoder knows all messages and side-information symbols, centralized coding allows us to achieve zero-error — for almost all realizations of (W_1, W_2, W_1', W_2) the Slepian-Wolf code works, and for the remaining ϵ-probable realizations, we simply send out (W_1, W_2), which has negligible impact on expected rate, as ϵ can be chosen to be arbitrarily small.

2.5.1 Class I: Linear Computation Broadcast

Because computations are often linear, it is of particular interest to consider the linear version of the computation broadcast problem, denoted linear computation broadcast, or LCB. For LCB, the defining restriction is that W_1, W_1', W_2, W_2' are arbitrary linear combinations of a basis set of independent symbols from a finite field. Let the basis symbols be specified through the $m \times 1$ column vector $\mathbf{X} = (x_1; x_2; \cdots; x_m)$, where $x_i, i \in \{1, 2, \cdots, m\}$ are i.i.d. uniform symbols from a finite field $\mathbb{F}_q$, $q = p^n$, for a prime p and an integer n. Since all symbols are linear combinations of the basis symbols, they are represented by $m \times 1$ vectors of linear combining coefficients. Each message or side-information is then specified in terms of such vectors,

$$\mathbf{W}_1 = \mathbf{X}^T\mathbf{V}_1 \quad (15)$$
$$\mathbf{W}_1' = \mathbf{X}^T\mathbf{V}_1' \quad (16)$$
$$\mathbf{W}_2 = \mathbf{X}^T\mathbf{V}_2 \quad (17)$$
$$\mathbf{W}_2' = \mathbf{X}^T\mathbf{V}_2' \quad (18)$$

For example, if $\mathbf{V}_1$ is comprised of two $m \times 1$ vectors, i.e., $\mathbf{V}_1 = [\mathbf{V}_1^{(1)}, \mathbf{V}_1^{(2)}]$, then it means that W_1 is comprised of symbols $\mathbf{X}^T\mathbf{V}_1^{(1)}, \mathbf{X}^T\mathbf{V}_2^{(2)}$, and may be represented as $\mathbf{W}_1 = [\mathbf{X}^T\mathbf{V}_1^{(1)}, \mathbf{X}^T\mathbf{V}_2^{(2)}]$. Note that the broadcast information S is *not* constrained to be a linear function of the basis symbols, although as we will prove, it turns out that linear forms of S are information theoretically optimal (refer to Section 5).

2.5.2 Class II: Matching Problems

While we are able to characterize the capacity of linear computation broadcast in this work, the capacity for non-linear settings remains open in general. In order to explore the challenges that arise in non-linear settings, we will focus on a limited class of non-linear computation broadcast problems, that we label as 'matching' problems. Here, the dependence between w_1 and w_2 is in the form of an invertible function (a perfect matching, equivalently a permutation) that depends upon w_1', w_2'. The dependence is minimal in the sense that each of (w_1', w_2', w_1) and (w_1', w_2', w_2) are independent and uniformly distributed over $[m_1] \times [m_2] \times [m]$. Mathematically,

$$(w_1, w_2, w_1', w_2') \in [m] \times [m] \times [m_1] \times [m_2], \quad (19)$$
$$H(w_1', w_2', w_1) = H(w_1') + H(w_2') + H(w_1) = \log_2(m_1) + \log_2(m_2) + \log_2(m), \quad (20)$$
$$H(w_1', w_2', w_2) = H(w_1') + H(w_2') + H(w_2) = \log_2(m_1) + \log_2(m_2) + \log_2(m), \quad (21)$$
$$H(w_1 \mid w_1', w_2', w_2) = H(w_2 \mid w_1', w_2', w_1) = 0. \quad (22)$$

Note that this setting includes both Example 1 and Example 2 as special cases when the matching is reduced to a linear mapping. We will explore how the *structure* of the matching affects the capacity of computation broadcast. In particular, we will characterize both minimally structured and maximally structured cases that correspond to the extremal values of capacity, while all other settings lie somewhere between these extremal values.

3 Results

3.1 A General Converse

Our first result is a general converse bound, stated in the following theorem.

Theorem 2 [General Converse] *For any computation broadcast problem,*

$$C_{CB} \;\leq\; \overline{C}_{CB} \;\leq\; \frac{H(w_1,w_2)}{H(w_1|w_1')+H(w_2|w_2')-\min\Big(I(w_1;w_2,w_2'|w_1'),I(w_2;w_1,w_1'|w_2')\Big)}. \tag{23}$$

The proof of Theorem 2 is presented in Section 4. In fact, the bound is intuitively quite obvious. The key to the bound is that

$$R_L^* \leq H(W_1,W_2)/\left[H(W_1|W_1')+H(W_2|W_2')-\min\Big(I(W_1;W_2,W_2'|W_1'),I(W_2;W_1,W_1'|W_2')\Big)\right]$$

which follows from the following two bounds.

$$H(S) \;\geq\; H(W_1 \mid W_1')+H(W_2 \mid W_1,W_1',W_2') \tag{24}$$
$$H(S) \;\geq\; H(W_2 \mid W_2')+H(W_1 \mid W_2,W_2',W_1') \tag{25}$$

For the first bound in (24), note that User 1, who already knows W_1', at least needs another $H(W_1|W_1')$ bits to decode W_1, and after everything known to User 1 is given to User 2 by a genie, User 2, who now knows W_1, W_1', W_2', needs another $H(W_2|W_1,W_1',W_2')$ bits to decode W_2. So without the genie we cannot need any less. The same intuition can be applied with the users switched for (25). In fact, the basic intuition is strong enough that the bound holds even in the relaxed entropic formulation, so we also have

$$\overline{C}_{CB} \leq H(w_1,w_2)/\left[H(w_1|w_1')+H(w_2|w_2')-\min\Big(I(w_1;w_2,w_2'|w_1'),I(w_2;w_1,w_1'|w_2')\Big)\right]$$

Finally, as discussed previously, $C_{CB} \leq \overline{C}_{CB}$ is true by definition since the entropic formulation is a relaxation of the complete (structural) formulation of the computation broadcast problem.

What is surprising about the converse bound is that it turns out to be tight for many settings of interest. In particular, for the linear computation broadcast problem, the converse bound is tight for both the entropic formulation as well as the structured formulation, i.e., it is also achievable. For the class of matching problems, the bound is tight for the entropic formulation, but not necessarily for the complete structured formulation, i.e., it is not achievable in general and the capacity may be strictly smaller once the dependency structure of the problem is fully accounted for. This makes sense because the converse bound is based on only entropic inequalities, in fact it uses only Shannon information inequalities, i.e., sub-modularity properties, so it cannot capture more structural constraints than the entropic formulation.

3.2 Capacity of Linear Computation Broadcast

Our second result shows that the bound in Theorem 2 is tight for the linear computation broadcast problem for any block length L. We state this result in the following theorem.

Theorem 3 *For linear computation broadcast, the capacity is*

$$C_{CB} \;=\; \overline{C}_{CB} \;=\; \frac{H(w_1,w_2)}{H(w_1|w_1')+H(w_2|w_2')-\min\Big(I(w_1;w_2,w_2'|w_1'),I(w_2;w_1,w_1'|w_2')\Big)}.$$

The proof of Theorem 3 is presented in Section 5. Since the converse is already available from Theorem 2, only a proof of achievability is needed. Intuitively, the achievable scheme is described as follows. First without loss of generality it is assumed that W_1 is independent of W_1', and similarly, W_2 is independent of W_2', because any dependence can be extracted separately as a sub-message that is already available to the user, and therefore can be eliminated from the user's demand. The core of the achievability argument then is that for linear computation broadcast, the problem can be partitioned into 3 independent sub-problems, labeled a, b, c. Correspondingly, each message is split into 3 independent parts: $\mathbf{W}_i = (\mathbf{W}_{ia}, \mathbf{W}_{ib}, \mathbf{W}_{ic})$, $i \in \{1, 2\}$. The 3 partitions are then solved as separate and independent problems, with corresponding solutions $\mathbf{S}_a, \mathbf{S}_b, \mathbf{S}_c$ that ultimately require a total of $H(S) = H(\mathbf{S}_a) + H(\mathbf{S}_b) + H(\mathbf{S}_c)$ bits. The sub-messages $\mathbf{W}_{1a}, \mathbf{W}_{2a}$ are analogous to Example 1, i.e., $\mathbf{W}_{1a}$ is a function[3] of $\mathbf{W}_2'$ while $\mathbf{W}_{2a}$ is a function of $\mathbf{W}_1'$, so that it suffices to send $H(\mathbf{S}_a) = \max(H(\mathbf{W}_{1a}), H(\mathbf{W}_{2a}))$ bits as in Example 1. The partition $\mathbf{W}_{1b}, \mathbf{W}_{2b}$ is analogous to Example 2, i.e., it satisfies a dependence relation of the form $\mathbf{W}_{1b}\mathbf{M}_{1b} + \mathbf{W}_{2b}\mathbf{M}_{2b} + \mathbf{W}_1'\mathbf{M}_1' + \mathbf{W}_2'\mathbf{M}_2' = \mathbf{0}$, where $H(\mathbf{W}_{1b}) = H(\mathbf{W}_{2b})$, $\mathbf{M}_1', \mathbf{M}_2', \mathbf{M}_{1b}, \mathbf{M}_{2b}$ are linear transformations (matrices) and $\mathbf{M}_{1b}, \mathbf{M}_{2b}$ are invertible. This is solved by sending, $\mathbf{S}_b = \mathbf{W}_{2b}\mathbf{M}_{2b} + \mathbf{W}_2'\mathbf{M}_2'$ which satisfies the demands of both users and requires $H(\mathbf{S}_b) = H(\mathbf{W}_{1b}) = H(\mathbf{W}_{2b})$ bits. Finally, the partition $\mathbf{W}_{1c}, \mathbf{W}_{2c}$ is trivial as it is comprised of sub-messages that are independent of each other and of all side-information, so the optimal solution for this part is simply uncoded transmission $\mathbf{S}_c = (\mathbf{W}_{1c}, \mathbf{W}_{2c})$ which takes $H(\mathbf{S}_c) = H(\mathbf{W}_{1c}) + H(\mathbf{W}_{2c})$ bits. Without loss of generality, suppose $H(\mathbf{W}_{1a}) \geq H(\mathbf{W}_{2a})$. Then, the total number of bits needed is $H(S) = H(\mathbf{S}_a) + H(\mathbf{S}_b) + H(\mathbf{S}_c) = H(\mathbf{W}_{1a}) + H(\mathbf{W}_{1b}) + H(\mathbf{W}_{1c}) + H(\mathbf{W}_{2c}) = H(\mathbf{W}_1 \mid \mathbf{W}_1') + H(\mathbf{W}_2 \mid \mathbf{W}_1, \mathbf{W}_1', \mathbf{W}_2')$ which matches the converse bound. Therefore $H(\mathbf{W}_1, \mathbf{W}_2)/C_{CB} = H(\mathbf{W}_1 \mid \mathbf{W}_1') + H(\mathbf{W}_2 \mid \mathbf{W}_1, \mathbf{W}_1', \mathbf{W}_2'))$ in this case. Note that if we assumed instead that $H(\mathbf{W}_{1a}) \leq H(\mathbf{W}_{2a})$ then the number of bits required by the achievable scheme, and the tight converse bound on $H(\mathbf{W}_1, \mathbf{W}_2)/C_{CB}$ (because it is achievable), would both be equal to $H(\mathbf{W}_2 \mid \mathbf{W}_2') + H(\mathbf{W}_1 \mid \mathbf{W}_2, \mathbf{W}_1', \mathbf{W}_2')$.

Example

Let $\mathbf{X} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T$, whose elements are i.i.d. uniform random variables in $\mathbb{F}_3$. Let us define

$$\mathbf{W}_1' = [x_1, x_3], \qquad \mathbf{W}_1 = [(x_1 + 2x_2),\ (x_3 + x_5),\ (x_1 + x_4 + x_6),\ x_7] \tag{26}$$

$$\mathbf{W}_2' = [x_2, x_4], \qquad \mathbf{W}_2 = [(2x_1 + x_2),\ x_5,\ (x_2 + x_4 + 2x_6)] \tag{27}$$

Splitting into a, b, c sub-problems (see Section 5), we have

$$\mathbf{W}_{1a} = [x_1 + 2x_2] \equiv [2x_2], \qquad \mathbf{W}_{1b} = [x_3 + x_5, x_1 + x_4 + x_6], \qquad \mathbf{W}_{1c} = [x_7] \tag{28}$$

$$\mathbf{W}_{2a} = [2x_1 + x_2] \equiv [2x_1], \qquad \mathbf{W}_{2b} = [x_5, x_2 + x_4 + 2x_6], \qquad \mathbf{W}_{2c} = [\ ] \tag{29}$$

Following the procedure in Section 5 we will find that $\mathbf{W}_{1a} = [x_1 + 2x_2]$, which makes $\mathbf{W}_{1a}$ a function of $(\mathbf{W}_1', \mathbf{W}_2')$. However, note that setting $\mathbf{W}_{1a} = [x_1 + 2x_2]$ is equivalent ('$\equiv$') to setting $\mathbf{W}_{1a} = [2x_2]$ because User 1 already knows x_1. In the same sense, setting $\mathbf{W}_{2a} = [2x_1 + x_2]$ is equivalent to setting $\mathbf{W}_{2a} = [2x_1]$ because x_2 is already known to User 2 as side-information. Thus, without loss of generality, $\mathbf{W}_{1a}$ is a function of only $\mathbf{W}_2'$, and $\mathbf{W}_{2a}$ is a function of only

[3] In fact $\mathbf{W}_{1a}$ may be a linear combination of both $\mathbf{W}_1', \mathbf{W}_2'$ (see (97)), but since $\mathbf{W}_1'$ is already known to User 1, there is no loss of generality in restricting $\mathbf{W}_{1a}$ to be the part that only depends on $\mathbf{W}_2'$. Similarly, there is no loss of generality in restricting $\mathbf{W}_{2a}$ to a function of $\mathbf{W}_1'$.

$\mathbf{W}'_1$. Thus, sub-problem 'a' is analogous to the setting of Example 1, and is solved by transmitting $\mathbf{S}_a = [2x_2 + 2x_1]$. For sub-problem '$b$', note that

$$\mathbf{W}_{1b}\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} + \mathbf{W}_{2b}\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} + \mathbf{W}'_1\begin{bmatrix} 0 & -2 \\ -1 & 0 \end{bmatrix} + \mathbf{W}'_2\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} = \mathbf{0} \tag{30}$$

and the matrices multiplying $\mathbf{W}_{1b}$ and $\mathbf{W}_{2b}$ are invertible matrices. This problem is analogous to Example 2 and is solved by sending $\mathbf{S}_b = \mathbf{W}_{2b}\mathbf{M}_{2b} + \mathbf{W}'_2\mathbf{M}'_2 = [-x_5, -2x_4 - 2x_6]$. Finally, sub-problem '$c$' is trivially solved by sending $\mathbf{S}_c = [\mathbf{W}_{1c}, \mathbf{W}_{2c}] = [x_7]$. Combining $\mathbf{S}_a, \mathbf{S}_b, \mathbf{S}_c$ into S, we have the solution,

$$S = ((2x_2 + 2x_1), (-x_5), (-2x_4 - 2x_6), (x_7)) \tag{31}$$

which needs $H(S) = 4$ symbols from $\mathbb{F}_3$ per block, and the rate achieved is $R = H(\mathbf{W}_1, \mathbf{W}_2)/H(S) = 7/4$. Since this matches the converse bound from Theorem 2, we have shown that for this example,

$$C_{CB} = 7/4. \tag{32}$$

3.3 Extra-entropic Structure Matters

Theorem 3 shows that the general converse of Theorem 2 is tight for linear computation broadcast, and the solution of the structural formulation in Section 2.1 coincides with the solution to the entropic formulation in Section 2.2, i.e., $C_{CB} = \overline{C}_{CB}$. Our next result shows that this is not the case in general.

Theorem 4 *There exist instances of the computation broadcast problem where $C_{CB} < \overline{C}_{CB}$. Thus, the converse in Theorem 2 is not always tight for the general (non-linear) computation broadcast problem, and extra-entropic structure matters.*

Proof: To prove this, we will present two instances of computation broadcast, say $\mathrm{CB}_1, \mathrm{CB}_2$, that have the same entropic formulations, so they have the same $\overline{C}_{CB}$. Yet, these two instances have different structural formulations that produce different capacities. Incidentally, both instances are matching problems.

CB_1: This instance of the computation broadcast problem is defined by $(w'_1, w'_2, w_1, w_2) \in \{0,1\} \times \{0,1\} \times \{0,1,2,3\} \times \{0,1,2,3\}$. The marginal distribution of each random variable is uniform over its own alphabet set. Furthermore, w'_1, w'_2, w_1 are independent and w_2 is uniquely determined by w'_1, w'_2, w_1 according to the functional relationship,

$$w_2 = (w_1 + z) \mod 4, \tag{33}$$

where z is a function of (w'_1, w'_2), defined as follows.

z	$w'_2 = 0$	$w'_2 = 1$
$w'_1 = 0$	0	1
$w'_1 = 1$	2	3

(34)

Thus, for all $w'_1, w'_2 \in \{0,1\}, w_1, w_2 \in \{0,1,2,3\}$

$$P_{w_1,w'_1,w_2,w'_2} = P_{w'_1}P_{w'_2}P_{w_1}P_{w_2|w'_1,w'_2,w_1} \tag{35}$$

$$= \quad 1/2 \times 1/2 \times 1/4 \times \mathbb{1}\Big(w_2 = (w_1 + z) \mod 4\Big) \tag{36}$$

where $\mathbb{1}(x)$ is the indicator function that takes value 1 if the event x is true and 0 otherwise. Note that given (w_1', w_2'), there is an invertible mapping between w_1 and w_2, which makes this a matching problem. The entropies of all subsets of $\{w_1', w_2', w_1, w_2\}$ are found as follows.

$$H(w_1') = H(w_2') = 1, \quad H(w_1) = H(w_2) = 2 \tag{37}$$

$$H(u, v) = H(u) + H(v), \ \forall \{u, v\} \subset \{w_1', w_2', w_1, w_2\} \tag{38}$$

$$H(t, u, v) = 4, \ \forall \{t, u, v\} \subset \{w_1', w_2', w_1, w_2\} \tag{39}$$

$$H(w_1, w_1', w_2, w_2') = 4 \tag{40}$$

Theorem 2 establishes a converse bound for this problem, $C_{CB} \leq \overline{C}_{CB} \leq 2$. The bound turns out to be achievable by setting $L = 1$ and choosing $S = (w_1 + 2w_1') \mod 4$ which satisfies both users' demands. This is verified as follows. User 1 obtains w_1 by computing $w_1 = (S - 2w_1') \mod 4$. User 2 obtains w_2 by computing $w_2 = (S + w_2') \mod 4$, which is possible because in this problem $z = (2w_1' + w_2') \mod 4$. Since $H(S) = 2$ bits and the rate achieved is $H(w_1, w_2)/H(S) = 2$, the achievability matches the converse, which proves that for CB_1, the capacity $C_{CB_1} = 2$.

CB_2: CB_2 is identical to CB_1 in all respects, except that the definition of z is slightly modified as follows.

z	$w_2' = 0$	$w_2' = 1$
$w_1' = 0$	0	1
$w_1' = 3$	3	2

(41)

The change in the z does not affect the entropic formulation of the problem. It is easily verified that the entropies of all subsets of $\{w_1', w_2', w_1, w_2\}$ are still given by (37)-(40). Since the entropic formulation is not affected we must still $\overline{C}_{CB_1} = \overline{C}_{CB_2} = 2$. However, the following lemma claims that the capacity $C_{CB_2} = \frac{4}{4 - \log_2(3)}$ is strictly smaller than C_{CB_1}, i.e., Theorem 4 is proved and the extra-entropic structure reduces capacity in this case.

Lemma 1 *For the computation broadcast problem CB_2 defined above,*

$$C_{CB_2} = \frac{4}{4 - \log_2(3)} \tag{42}$$

The proof of Lemma 1 is presented in Section 6. ■

3.4 Capacity of Matching Computation Broadcast

To gain a deeper understanding of the significance of extra-entropic structure that is revealed by CB_1 and CB_2, we explore the capacity of a class of computation broadcast problems called matching problems, which include CB_1 and CB_2 as special cases. For matching problems we have $(w_1, w_2, w_1', w_2') \in [m_1] \times [m_2] \times [m] \times [m]$ where $m_1, m_2, m \in \mathbb{N}$. The tuple (w_1', w_2', w_1) is uniformly distributed over $[m_1] \times [m_2] \times [m]$, while w_2 is a function of w_1', w_2', w_1 defined as,

$$w_2 \quad = \quad \pi_{w_1', w_2'}(w_1) \tag{43}$$

where $\pi_{w_1',w_2'}$ is a permutation on $[m]$ that depends on the realization of the side-information (w_1', w_2'). Distinct realizations of (w_1', w_2') may or may not produce distinct permutations. $\pi_{w_1',w_2'}$ may be represented in a matrix form as follows.

$\pi_{w_1',w_2'}$	$w_2'=1$	$w_2'=2$	$\cdots$	$w_2'=m_2$
$w_1'=1$	$\pi_{1,1}$	$\pi_{1,2}$	$\cdots$	π_{1,m_2}
$w_1'=2$	$\pi_{2,1}$	$\pi_{2,2}$	$\cdots$	π_{2,m_2}
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$w_1'=m_1$	$\pi_{m_1,1}$	$\pi_{m_1,2}$	$\cdots$	π_{m_1,m_2}

Let this matrix be denoted by Π. Specification of Π completely defines the structure of the matching computation broadcast problem. For all $w_1' \in [m_1], w_2' \in [m_2], w_1, w_2 \in [m]$, we have

$$\begin{aligned} P_{w_1,w_1',w_2,w_2'} &= P_{w_1'} P_{w_2'} P_{w_1} P_{w_2|w_1',w_2',w_1} && (44) \\ &= 1/m_1 \times 1/m_2 \times 1/m \times \mathbb{1}(w_2 = \pi_{w_1',w_2'}(w_1)) && (45) \end{aligned}$$

Note that w_1', w_2', w_2 are independent.

Next let us introduce some definitions that are useful to gauge the amount of structure in a given Π. We begin with the notion of a cycle, which is a closed path on an $m_1 \times m_2$ grid, obtained by a sequence of alternating horizontal and vertical steps. See Fig. 2 for an illustration.

Definition 1 (Cycle) *Let $N \geq 4$ be an even number. We say that the N terms, (a_1, b_1), (a_2, b_2), $\cdots$, $(a_N, b_N) \in [m_1] \times [m_2]$, form a cycle of length N in $[m_1] \times [m_2]$, denoted by*

$$(a_1, b_1) \leftrightarrow (a_2, b_2) \leftrightarrow \cdots \leftrightarrow (a_N, b_N) \leftrightarrow (a_1, b_1) \tag{46}$$

if both of the following properties are true $\forall i \in [N]$:

1. *$a_i = a_{i+1}$ and $b_i \neq b_{i+1}$ if i is odd.*
2. *$b_i = b_{i+1}$ and $a_i \neq a_{i+1}$ if i is even.*

where we interpret all indices modulo N (so, e.g., $a_{N+1} = a_1$).

Other descriptions are also possible for the same cycle. For example, the cycle in Fig. 2 can also be identified as $(5,2) \leftrightarrow (5,5) \leftrightarrow (4,5) \leftrightarrow (4,3) \leftrightarrow (3,3) \leftrightarrow (3,1) \leftrightarrow (1,1) \leftrightarrow (1,2) \leftrightarrow (5,2)$.

Definition 2 (Induced Permutation) *For a cycle $(a_i, b_i)_{i \in [N]}$, we define its induced permutation as*

$$\pi_{a_1,b_1} \pi^{-1}_{a_2,b_2} \pi_{a_3,b_3} \pi^{-1}_{a_4,b_4} \cdots \pi^{-1}_{a_N,b_N} \tag{47}$$

Definition 3 (Maximally Structured) *We say that Π is maximally structured if the induced permutation for every possible cycle in $[m_1] \times [m_2]$ is the identity.*[4]

Definition 4 (Minimally Structured) *We say that Π is minimally structured if the induced permutation for every possible cycle in $[m_1] \times [m_2]$ is a derangement.*[5]

[4] A permutation π on $[m]$ is the identity if and only if it maps every element to itself, i.e., $\pi[i] = i$ for all $i \in [m]$.

[5] A permutation π on $[m]$ is a derangement if and only if no element is mapped to itself, i.e., $\pi[i] \neq i$ for all $i \in [m]$.

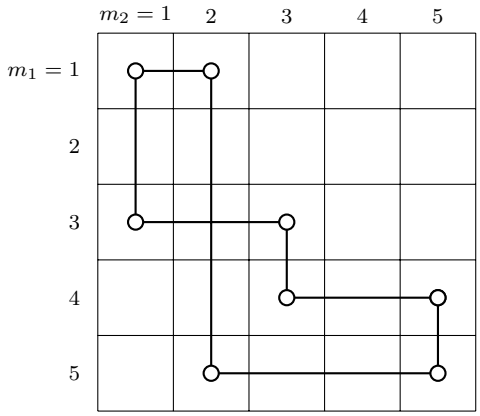


Figure 2: A cycle, $(1,1) \leftrightarrow (1,2) \leftrightarrow (5,2) \leftrightarrow (5,5) \leftrightarrow (4,5) \leftrightarrow (4,3) \leftrightarrow (3,3) \leftrightarrow (3,1) \leftrightarrow (1,1)$.

Maximal structure is a generalization of the setting in CB_1. In CB_1 there is only one possible cycle: $(1,1) \leftrightarrow (1,2) \leftrightarrow (2,2) \leftrightarrow (2,1) \leftrightarrow (1,1)$, for which the induced permutation $\pi_{1,1}\pi_{1,2}^{-1}\pi_{2,2}\pi_{2,1}^{-1}$ is the identity. Minimal structure is a generalization of the setting in CB_2. For the cycle $(1,1) \leftrightarrow (1,2) \leftrightarrow (2,2) \leftrightarrow (2,1) \leftrightarrow (1,1)$, the induced permutation $\pi_{1,1}\pi_{1,2}^{-1}\pi_{2,2}\pi_{2,1}^{-1}$ is a derangement.

The significance of this structure is revealed by the next theorem.

Theorem 5 *For a matching computation broadcast problem specified by the structure* Π*,*

$$\frac{2\log_2(m)}{\log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)} \leq C_{CB} \leq 2.$$

The upper bound is tight if Π *is maximally structured. The lower bound is tight if* Π *is minimally structured.*

The proof of Theorem 5 is presented in Section 7. The following observations are in order.

1. Since maximally structured settings represent the best case and minimally structured settings the worst case, it is evident that structure is beneficial.

2. The proof presented in Section 7 shows that the minimally structured setting still has some (unavoidable) combinatoric structure that is critical for the optimal achievable scheme.

3. To contrast with the previous observation, consider the following. Suppose $m_1 = m_2 \triangleq m'$ and all alphabet sizes grow together proportionately. Then the minimally structured setting essentially loses all its structure and random binning is close to optimal. To see this, consider the term $\log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)$. For large values of m', this becomes $\approx 2\log_2(m') - \log_2(2m') = \log_2(m') - 1 = H(w_1') - 1$. So the capacity C_{CB} approaches the value $H(w_1, w_2)/[H(w_1) + H(w_1')]$ which is achievable[6] by random binning. Thus, random binning is asymptotically optimal for minimally structured instances of matching computation broadcast.

[6]It is achieved by separately compressing and sending w_1, w_1'. User 1 directly receives w_1 and User 2 decodes $w_2 = \pi_{w_1', w_2'}(w_1)$.

4 Proof of Theorem 2: A General Converse

The converse in Theorem 2 consists of the following two bounds.

$$\begin{aligned} H(S) &\geq H(W_1|W_1') + H(W_2|W_2') - I(W_2; W_1, W_1'|W_2') && (48)\\ &= H(W_1|W_1') + H(W_2|W_1, W_1', W_2') && (49)\\ H(S) &\geq H(W_2|W_2') + H(W_1|W_2, W_2', W_1') && (50) \end{aligned}$$

We only need to prove (49), as the proof of (50) follows from symmetry. The proof of (49) is presented next. Note that in the proofs, the relevant equations needed to justify each step are specified by the equation numbers set on top of the (in)equality symbols.

We expand the joint entropy $H(S, W_1|W_1')$ in two different ways. On the one hand, we have

$$\begin{aligned} H(S, W_1|W_1') &= H(S|W_1') + H(W_1|W_1', S) && (51)\\ &\overset{(1)}{\leq} H(S) && (52) \end{aligned}$$

On the other hand, we have

$$\begin{aligned} & H(S, W_1|W_1') && (53)\\ = \; & H(W_1|W_1') + H(S|W_1, W_1') && (54)\\ \geq \; & H(W_1|W_1') + H(S|W_1, W_1') - H(S|W_1, W_1', W_2, W_2') && (55)\\ = \; & H(W_1|W_1') + I(S; W_2, W_2'|W_1, W_1') && (56)\\ = \; & H(W_1|W_1') + H(W_2, W_2'|W_1, W_1') - H(W_2, W_2'|W_1, W_1', S) && (57)\\ = \; & H(W_1|W_1') + H(W_2'|W_1, W_1') + H(W_2|W_1, W_1', W_2') - H(W_2'|W_1, W_1', S) \\ & - H(W_2|W_1, W_1', W_2', S) && (58)\\ \overset{(2)}{=} \; & H(W_1|W_1') + H(W_2|W_1, W_1', W_2') + I(S; W_2'|W_1, W_1') && (59)\\ \geq \; & H(W_1|W_1') + H(W_2|W_1, W_1', W_2') && (60) \end{aligned}$$

Thus combining (52) and (60), we have the desired bound (49). The proof of Theorem 2 is complete.

5 Proof of Theorem 3: Linear Achievability

Without loss of generality we will assume that W_1 is independent of W_1', and similarly, W_2 is independent of W_2'. There is no loss of generality in this assumption because any linear dependence between W_1 and W_1', or between W_2 and W_2', can be extracted separately as a sub-message that is already available to the user, and therefore can be eliminated from the user's demand.

Recall that $\mathbf{X} = (x_1; x_2; \ldots; x_m)$ is an $m \times 1$ random vector, whose elements x_i are i.i.d. uniform over $\mathbb{F}_q$. All entropies in this section are measured in units of q-ary symbols. For any matrix $\mathbf{A} \in \mathbb{F}_q^{m \times n}$, we will use the notation A to denote the set of column vectors of $\mathbf{A}$.

Lemma 2 *For an arbitrary $m \times n$ matrix $\mathbf{A} \in \mathbb{F}_q^{m \times n}$, $H(\mathbf{X}^T\mathbf{A}) = rank(\mathbf{A})$.*

Definition 5 (Independent subspaces) *Subspaces $\mathcal{A}, \mathcal{B} \subset \mathbb{F}_q^m$ are independent if $\mathcal{A} \cap \mathcal{B} = \{\mathbf{0}\}$.*

Lemma 3 *For arbitrary matrices* $\mathbf{A} \in \mathbb{F}_q^{m \times n_A}, \mathbf{B} \in \mathbb{F}_q^{m \times n_B}$*, the mutual information* $I(\mathbf{X}^T\mathbf{A}; \mathbf{X}^T\mathbf{B}) = 0$ *if and only if span*(A) *and span*(B) *are independent subspaces.*

The proofs of Lemma 2 and Lemma 3 are immediate and are deferred to the Appendix.
Define

$$\mathbf{V}_{1a} = \left[\mathbf{V}_{1a}^{(1)}, \mathbf{V}_{1a}^{(2)}, \cdots, \mathbf{V}_{1a}^{(n_{1a})}\right] \in \mathbb{F}_q^{m \times n_{1a}}, \quad \mathbf{V}_{2a} = \left[\mathbf{V}_{2a}^{(1)}, \mathbf{V}_{2a}^{(2)}, \cdots, \mathbf{V}_{2a}^{(n_{2a})}\right] \in \mathbb{F}_q^{m \times n_{2a}} \tag{61}$$

$$\mathbf{V}_{1b} = \left[\mathbf{V}_{1b}^{(1)}, \mathbf{V}_{1b}^{(2)}, \cdots, \mathbf{V}_{1b}^{(n_{1b})}\right] \in \mathbb{F}_q^{m \times n_{1b}}, \quad \mathbf{V}_{2b} = \left[\mathbf{V}_{2b}^{(1)}, \mathbf{V}_{2b}^{(2)}, \cdots, \mathbf{V}_{2b}^{(n_{2b})}\right] \in \mathbb{F}_q^{m \times n_{2b}} \tag{62}$$

$$\mathbf{V}_{1c} = \left[\mathbf{V}_{1c}^{(1)}, \mathbf{V}_{1c}^{(2)}, \cdots, \mathbf{V}_{1c}^{(n_{1c})}\right] \in \mathbb{F}_q^{m \times n_{1c}}, \quad \mathbf{V}_{2c} = \left[\mathbf{V}_{2c}^{(1)}, \mathbf{V}_{2c}^{(2)}, \cdots, \mathbf{V}_{2c}^{(n_{2c})}\right] \in \mathbb{F}_q^{m \times n_{2c}} \tag{63}$$

such that

1. V_{1a}, V_{1b}, V_{1c} are disjoint sets.
2. V_{1a} is a basis for $\text{span}(V_1) \cap \text{span}(V_1' \cup V_2')$.
3. $V_{1a} \cup V_{1b}$ is a basis for $\text{span}(V_1) \cap \text{span}(V_1' \cup V_2' \cup V_2)$.
4. $V_{1a} \cup V_{1b} \cup V_{1c}$ is a basis for $\text{span}(V_1)$.
5. V_{2a}, V_{2b}, V_{2c} are disjoint sets.
6. V_{2a} is a basis for $\text{span}(V_2) \cap \text{span}(V_1' \cup V_2')$.
7. $V_{2a} \cup V_{2b}$ is a basis for $\text{span}(V_2) \cap \text{span}(V_1' \cup V_2' \cup V_1)$.
8. $V_{2a} \cup V_{2b} \cup V_{2c}$ is a basis for $\text{span}(V_2)$.

Recall that basis vectors must be linearly independent. The existence of such V_{ia}, V_{ib}, V_{ic}, $i \in \{1, 2\}$, follows from Steinitz exchange lemma which guarantees that given a set of basis vectors $\{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_k\}$ for a k-dimensional subspace $\mathcal{P}$, and an arbitrary m-dimensional vector space $\mathcal{Q}$, such that $\mathcal{P} \subset \mathcal{Q}$, there exist $\mathbf{q}_1, \cdots, \mathbf{q}_{m-k} \in \mathcal{Q} \backslash \mathcal{P}$ such that $\{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_k, \mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{m-k}\}$ is a basis for $\mathcal{Q}$.

Remark: As an illustration of this construction, consider the example presented in Section 3.2 where we have,

$$\mathbf{X} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T \tag{64}$$

$$\mathbf{W}_1' = [x_1, x_3], \ \mathbf{W}_1 = [x_1 + 2x_2, x_3 + x_5, x_1 + x_4 + x_6, x_7] \tag{65}$$

$$\mathbf{W}_2' = [x_2, x_4], \ \mathbf{W}_2 = [2x_1 + x_2, x_5, x_2 + x_4 + 2x_6] \tag{66}$$

This gives us,

$$\mathbf{V}_1' = [\mathbf{e}_1, \mathbf{e}_3], \qquad \mathbf{V}_1 = [\mathbf{e}_1 + 2\mathbf{e}_2, \mathbf{e}_3 + \mathbf{e}_5, \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_6, \mathbf{e}_7] \tag{67}$$

$$\mathbf{V}_2' = [\mathbf{e}_2, \mathbf{e}_4], \qquad \mathbf{V}_2 = [2\mathbf{e}_1 + \mathbf{e}_2, \mathbf{e}_5, \mathbf{e}_2 + \mathbf{e}_4 + 2\mathbf{e}_6] \tag{68}$$

and

$$\mathbf{V}_{1a} = [\mathbf{e}_1 + 2\mathbf{e}_2], \qquad \mathbf{V}_{2a} = [2\mathbf{e}_1 + \mathbf{e}_2] \tag{69}$$

$$\mathbf{V}_{1b} = [\mathbf{e}_3 + \mathbf{e}_5, \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_6], \qquad \mathbf{V}_{2b} = [\mathbf{e}_5, \mathbf{e}_2 + \mathbf{e}_4 + 2\mathbf{e}_6] \tag{70}$$

$$\mathbf{V}_{1c} = [\mathbf{e}_7], \qquad \mathbf{V}_{2c} = [\,] \tag{71}$$

where $\mathbf{e}_i$ *denotes the* i^{th} *column of the* 7×7 *identity matrix.*

Next, for $i \in \{1,2\}$ and $\{i, i^c\} = \{1,2\}$, define $\mathbf{W}_{ia} = \mathbf{X}^T\mathbf{V}_{ia}$, $\mathbf{W}_{ib} = \mathbf{X}^T\mathbf{V}_{ib}$, $\mathbf{W}_{ic} = \mathbf{X}^T\mathbf{V}_{ic}$, so that

$$H(\mathbf{W}_{ia}, \mathbf{W}_{ib}, \mathbf{W}_{ic}) = H(\mathbf{W}_i) \tag{72}$$
$$H(\mathbf{W}_{ia}) + H(\mathbf{W}_{ib}) + H(\mathbf{W}_{ic}) = H(\mathbf{W}_i) \tag{73}$$
$$n_{ia} + n_{ib} + n_{ic} = H(\mathbf{W}_i) \tag{74}$$
$$H(\mathbf{W}_{ia}) = n_{ia} \tag{75}$$
$$H(\mathbf{W}_{ia} \mid \mathbf{W}'_1, \mathbf{W}'_2) = 0 \tag{76}$$
$$H(\mathbf{W}_{ib} \mid \mathbf{W}'_1, \mathbf{W}'_2) = n_{ib} \tag{77}$$
$$H(\mathbf{W}_{ia}, \mathbf{W}_{ib} \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_{i^c}) = 0 \tag{78}$$
$$H(\mathbf{W}_{ic} \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_{i^c}) = n_{ic} \tag{79}$$
$$H(\mathbf{W}_{ia}, \mathbf{W}_{ib}, \mathbf{W}_{ic} \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_1, \mathbf{W}_2) = 0 \tag{80}$$

(72) follows from the fact that $V_{ia} \cup V_{ib} \cup V_{ic}$ is the basis for the space spanned by V_i, which makes $\mathbf{W}_i$ an invertible function of $(\mathbf{W}_{ia}, \mathbf{W}_{ib}, \mathbf{W}_{ic})$. (73)-(75) follow from Lemma 2 and the fact that the $n_{ia}+n_{ib}+n_{ic}$ vectors in $V_{ia} \cup V_{ib} \cup V_{ic}$ form a basis, so they are linearly independent. (76) holds because $V_{ia} \subset \text{span}(V'_1 \cup V'_2)$, which makes $\mathbf{W}_{ia}$ a function of $\mathbf{W}'_1, \mathbf{W}'_2$. (77) holds because $\text{span}(V_{ib})$ is independent of $\text{span}(V'_1 \cup V'_2)$. This is because if a non-zero vector $\mathbf{U} \in \text{span}(V_{ib}) \cap \text{span}(V'_1 \cup V'_2)$ then $\mathbf{U} \in \text{span}(V_{ib})$ and $\mathbf{U} \in \text{span}(V_{ia})$, i.e., V_{ib} and V_{ia} do not span independent spaces, so $V_{ia} \cup V_{ib} \cup V_{ic}$ cannot be a set of basis vectors. Similarly, (79) holds because V_{ic} and $V'_1 \cup V'_2 \cup V_{i^c}$ span independent spaces (otherwise V_{ic} and V_{ib} cannot span independent spaces). (78) holds because $V_{ia} \cup V_{ib} \subset \text{span}(V'_1 \cup V'_2 \cup V_{i^c})$, and (80) holds because $V_{ia} \cup V_{ib} \cup V_{ic} \subset \text{span}(V'_1 \cup V'_2 \cup V_1 \cup V_2)$.

Since $V_{1b} \subset \text{span}(V'_1 \cup V'_2 \cup V_2)$, there exist matrices $\mathbf{M}'_1 \in \mathbb{F}_q^{n'_1 \times n_{1b}}$, $\mathbf{M}'_2 \in \mathbb{F}_q^{n'_2 \times n_{1b}}$, $\mathbf{M}_{2a} \in \mathbb{F}_q^{n_{2a} \times n_{1b}}$, $\mathbf{M}_{2b} \in \mathbb{F}_q^{n_{2b} \times n_{1b}}$, $\mathbf{M}_{2c} \in \mathbb{F}_q^{n_{2c} \times n_{1b}}$, such that

$$\mathbf{V}_{1b} = \mathbf{V}'_1\mathbf{M}'_1 + \mathbf{V}'_2\mathbf{M}'_2 + \mathbf{V}_{2a}\mathbf{M}_{2a} + \mathbf{V}_{2b}\mathbf{M}_{2b} + \mathbf{V}_{2c}\mathbf{M}_{2c}. \tag{81}$$

We will now show that without loss of generality, $\mathbf{M}_{2a}, \mathbf{M}_{2c}$ are zero matrices, and $\mathbf{M}_{2b}$ is an invertible square matrix. Since $\mathbf{V}_{2a}$ can be expanded as a linear combination of $\mathbf{V}'_1$ and $\mathbf{V}'_2$, and absorbed into corresponding terms in (81), there is no loss of generality in the assumption that $\mathbf{M}_{2a}$ is the zero matrix, i.e., a matrix whose elements are all zeros. Next, without loss of generality, we can also assume $\mathbf{M}_{2c}$ is a zero matrix because $\text{span}(V_{2c})$ and $\text{span}(V'_1 \cup V'_2 \cup V_{2b} \cup V_{1b})$ are independent subspaces. This is because of Lemma 3 and the fact that $\mathbf{W}_{2c}$ is independent of $(\mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_{2b}, \mathbf{W}_{1b})$ as shown below.

$$I(\mathbf{W}_{2c}; \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_{2b}, \mathbf{W}_{1b}) = H(\mathbf{W}_{2c}) - H(\mathbf{W}_{2c} \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_{2b}, \mathbf{W}_{1b}) \tag{82}$$
$$= n_{2c} - H(\mathbf{W}_{2c} \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_{2b}, \mathbf{W}_{1b}) \tag{83}$$
$$\leq n_{2c} - H(\mathbf{W}_{2c} \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_{2b}, \mathbf{W}_1) \tag{84}$$
$$= n_{2c} - H(\mathbf{W}_{2c} \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_1) \tag{85}$$
$$= 0. \tag{86}$$

(84) holds because $V_{1b} \subset \text{span}(V_1)$, which makes $\mathbf{W}_{1b}$ a function of $\mathbf{W}_1$, while (85) holds because according to (78) $\mathbf{W}_{2b}$ is a function of $\mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_1$. Thus, without loss of generality (81) reduces

to

$$\mathbf{V}_{1b} = \mathbf{V}'_1\mathbf{M}'_1 + \mathbf{V}'_2\mathbf{M}'_2 + \mathbf{V}_{2b}\mathbf{M}_{2b}. \tag{87}$$

Next, let us prove that $\mathbf{M}_{2b}$ is a square matrix, i.e., $n_{1b} = n_{2b}$.

$$\begin{aligned} I(\mathbf{W}_1; \mathbf{W}_2 \mid \mathbf{W}'_1, \mathbf{W}'_2) &= H(\mathbf{W}_1 \mid \mathbf{W}'_1, \mathbf{W}'_2) - H(\mathbf{W}_1 \mid \mathbf{W}_2, \mathbf{W}'_1, \mathbf{W}'_2) && (88)\\ &= n_{1b} + n_{1c} - n_{1c} && (89)\\ &= n_{1b} && (90) \end{aligned}$$

and similarly,

$$\begin{aligned} I(\mathbf{W}_1; \mathbf{W}_2 \mid \mathbf{W}'_1, \mathbf{W}'_2) &= H(\mathbf{W}_2 \mid \mathbf{W}'_1, \mathbf{W}'_2) - H(\mathbf{W}_2 \mid \mathbf{W}_1, \mathbf{W}'_1, \mathbf{W}'_2) && (91)\\ &= n_{2b} + n_{2c} - n_{2c} && (92)\\ &= n_{2b} && (93) \end{aligned}$$

Therefore, $n_{1b} = n_{2b} \triangleq n_b$ and $\mathbf{M}_{2b}$ is a square matrix. Next, let us prove that it has full rank. Suppose on the contrary that $\mathbf{M}_{2b}\mathbf{U} = \mathbf{0}$ for some $\mathbf{U} \in \mathbb{F}_q^{n_b \times 1}$ which is not the zero vector. Then (81) implies that $\mathbf{V}_{1b}\mathbf{U} = \mathbf{V}'_1\mathbf{M}'_1\mathbf{U} + \mathbf{V}'_2\mathbf{M}'_2\mathbf{U} \in \text{span}(V_{1a})$. But $\mathbf{V}_{1b}\mathbf{U}$ also belongs to $\text{span}(V_{1b})$. Since $\text{span}(V_{1a})$ and $\text{span}(V_{1b})$ are independent subspaces, we must have $\mathbf{V}_{1b}\mathbf{U} = \mathbf{0}$. This is a contradiction because V_{1b} is comprised of linearly independent vectors (because it is a basis), and $\mathbf{U}$ is not the zero vector. The contradiction proves that $\mathbf{M}_{2b}$ must have full rank, i.e., it must be invertible.

Remark: For the example presented in Section 3.2 and matrices specified in (67) - (71), we have

$$\begin{aligned} \mathbf{V}_{1b} &= [\mathbf{e}_3 + \mathbf{e}_5, \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_6] && (94)\\ &= [\mathbf{e}_1, \mathbf{e}_3] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + [\mathbf{e}_2, \mathbf{e}_4] \begin{bmatrix} 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} + [\mathbf{e}_5, \mathbf{e}_2 + \mathbf{e}_4 + 2\mathbf{e}_6] \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} && (95)\\ &= \mathbf{V}'_1\mathbf{M}'_1 + \mathbf{V}'_2\mathbf{M}'_2 + \mathbf{V}_{2b}\mathbf{M}_{2b} && (96) \end{aligned}$$

Without loss of generality, suppose $n_{1a} \geq n_{2a}$. Since $V_{1a} \subset \text{span}(V'_1 \cup V'_2)$, there exist $\mathbf{P}'_1 \in \mathbb{F}_q^{n'_1 \times n_{1a}}$, $\mathbf{P}'_2 \in \mathbb{F}_q^{n'_2 \times n_{1a}}$, such that the $m \times n_{1a}$ matrix

$$\mathbf{V}_{1a} = \mathbf{V}'_1\mathbf{P}'_1 + \mathbf{V}'_2\mathbf{P}'_2. \tag{97}$$

Similarly, there exist $\mathbf{Q}'_1 \in \mathbb{F}_q^{n'_1 \times n_{1a}}$, $\mathbf{Q}'_2 \in \mathbb{F}_q^{n'_2 \times n_{1a}}$, such that the $m \times n_{1a}$ matrix

$$\left[\mathbf{V}_{2a}, \mathbf{0}_{m \times (n_{1a} - n_{2a})}\right] = \mathbf{V}'_1\mathbf{Q}'_1 + \mathbf{V}'_2\mathbf{Q}'_2. \tag{98}$$

Note that $n_{1a} - n_{2a}$ columns of zeros are appended to $\mathbf{V}_{2a}$ to create a matrix the same size as $\mathbf{V}_{1a}$.

The transmitted vector $\mathbf{S} \in \mathbb{F}_q^{(n_{1a}+n_{1b}+n_{1c}+n_{2c}) \times 1}$ is now specified as

$$\mathbf{S} = \left(\underbrace{\mathbf{X}^T(\mathbf{V}'_1\mathbf{Q}'_1 + \mathbf{V}'_2\mathbf{P}'_2)}_{\mathbf{S}_a:\ 1 \times n_{1a}},\ \underbrace{\mathbf{X}^T(\mathbf{V}_{2b}\mathbf{M}_{2b} + \mathbf{V}'_2\mathbf{M}'_2)}_{\mathbf{S}_b:\ 1 \times n_{1b}},\ \underbrace{\mathbf{X}^T\mathbf{V}_{1c}, \mathbf{X}^T\mathbf{V}_{2c}}_{\mathbf{S}_c:\ (1 \times n_{1c}), (1 \times n_{2c})} \right)^T \tag{99}$$

Remark: For the example presented in Section 3.2 and matrices specified in (67) - (71), we have

$$\mathbf{V}_{1a} = [\mathbf{e}_1 + 2\mathbf{e}_2] = [\mathbf{e}_1, \mathbf{e_3}] \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [\mathbf{e}_2, \mathbf{e}_4] \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \mathbf{V}'_1\mathbf{P}'_1 + \mathbf{V}'_2\mathbf{P}'_2 \tag{100}$$

$$\mathbf{V}_{2a} = [2\mathbf{e}_1 + \mathbf{e}_2] = [\mathbf{e}_1, \mathbf{e_3}] \begin{bmatrix} 2 \\ 0 \end{bmatrix} + [\mathbf{e}_2, \mathbf{e}_4] \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mathbf{V}'_1\mathbf{Q}'_1 + \mathbf{V}'_2\mathbf{Q}'_2 \tag{101}$$

$$\mathbf{S}_a = \mathbf{X}^T(\mathbf{V}'_1\mathbf{Q}'_1 + \mathbf{V}'_2\mathbf{P}'_2) = \mathbf{X}^T(2\mathbf{e}_1 + 2\mathbf{e}_2) = 2x_1 + 2x_2 \tag{102}$$

$$\mathbf{S}_b = \mathbf{X}^T(\mathbf{V}_{2b}\mathbf{M}_{2b} + \mathbf{V}'_2\mathbf{M}'_2) = \mathbf{X}^T(\mathbf{e}_5, \mathbf{e}_4 + \mathbf{e}_6) = (x_5, x_4 + x_6) \tag{103}$$

$$\mathbf{S}_c = (\mathbf{X}^T\mathbf{V}_{1c}, \mathbf{X}^T\mathbf{V}_{2c}) = \mathbf{X}^T\mathbf{e}_7 = x_7 \tag{104}$$

Note that $\mathbf{S}_b$ *in (103) is slightly different (in fact invertible) from that in (31) because here the invertible matrix* $\mathbf{M}_{1b}$ *is absorbed in* $\mathbf{M}_{2b}$*.*

Let us verify that each user can recover their desired message from $\mathbf{S}$ and their own side-information.

$$\mathbf{S}_a - \mathbf{W}'_1\mathbf{Q}'_1 + \mathbf{W}'_1\mathbf{P}'_1 = \mathbf{X}^T(\mathbf{V}'_1\mathbf{Q}'_1 + \mathbf{V}'_2\mathbf{P}'_2 - \mathbf{V}'_1\mathbf{Q}'_1 + \mathbf{V}'_1\mathbf{P}'_1) = \mathbf{X}^T\mathbf{V}_{1a} = \mathbf{W}_{1a} \tag{105}$$

$$\mathbf{S}_b + \mathbf{W}'_1\mathbf{M}'_1 = \mathbf{X}^T(\mathbf{V}_{2b}\mathbf{M}_{2b} + \mathbf{V}'_2\mathbf{M}'_2 + \mathbf{V}'_1\mathbf{M}'_1) = \mathbf{X}^T\mathbf{V}_{1b} = \mathbf{W}_{1b} \tag{106}$$

$$\mathbf{S}_a - \mathbf{W}'_2\mathbf{P}'_2 + \mathbf{W}'_2\mathbf{Q}'_2 = \mathbf{X}^T(\mathbf{V}'_1\mathbf{Q}'_1 + \mathbf{V}'_2\mathbf{P}'_2 - \mathbf{V}'_2\mathbf{P}'_2 + \mathbf{V}'_2\mathbf{Q}'_2) = \mathbf{X}^T[\mathbf{V}_{2a}, \mathbf{0}] = [\mathbf{W}_{2a}, \mathbf{0}] \tag{107}$$

$$(\mathbf{S}_b - \mathbf{W}'_2\mathbf{M}'_2)\mathbf{M}_{2b}^{-1} = \mathbf{X}^T(\mathbf{V}_{2b}\mathbf{M}_{2b} + \mathbf{V}'_2\mathbf{M}'_2 - \mathbf{V}'_2\mathbf{M}'_2)\mathbf{M}_{2b}^{-1} = \mathbf{X}^T\mathbf{V}_{2b} = \mathbf{W}_{2b} \tag{108}$$

$$\mathbf{S}_c = (\mathbf{W}_{1c}, \mathbf{W}_{2c}) \tag{109}$$

Thus, User 1 is able to recover $\mathbf{W}_{1a}$ from $(\mathbf{S}_a, \mathbf{W}'_1)$ according to (105), $\mathbf{W}_{1b}$ from $(\mathbf{S}_b, \mathbf{W}'_1)$ according to (106), and $\mathbf{W}_{1c}$ directly from $\mathbf{S}_c$. Similarly, User 2 is able to recover $\mathbf{W}_{2a}$ from $(\mathbf{S}_a, \mathbf{W}'_2)$ according to (107), $\mathbf{W}_{2b}$ from $(\mathbf{S}_b, \mathbf{W}'_2)$ according to (108), and $\mathbf{W}_{2c}$ directly from $\mathbf{S}_c$.

Finally, note that $H(\mathbf{S}) \leq n_{1a} + n_{1b} + n_{1c} + n_{2c} = H(\mathbf{W}_1) + H(\mathbf{W}_2 \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_1) = H(\mathbf{W}_1 \mid \mathbf{W}'_1) + H(\mathbf{W}_2 \mid \mathbf{W}'_1, \mathbf{W}'_2, \mathbf{W}_1)$ which matches the converse. ■

6 Proof of Lemma 1

Define the optimal normalized broadcast cost as

$$H^* \triangleq \frac{H(w_1, w_2)}{C_{CB}} = \inf \frac{H(S)}{L}. \tag{110}$$

Note that the infimum is over all feasible S subject to (1), (2), (3) and $L \in \mathbb{N}$. Now proving $C_{CB_2} = 4/(4 - \log_2(3))$ is equivalent to proving that $H^* = 4 - \log_2(3)$. To show that $H^* = 4 - \log_2(3)$ bits, we first prove a converse bound that shows $H^* \geq 4 - \log_2(3)$ in Section 6.1, and then an achievable scheme that shows $H^* \leq 4 - \log_2(3)$ in Section 6.2.

6.1 Converse: $H^* \geq 4 - \log_2(3)$ bits

Let us start with a key observation, stated in the following lemma.

Lemma 4 *Any achievable scheme for* CB_2 *with block length* L*, i.e., any* $P_{S|W_1,W_2,W'_1,W'_2}$ *that satisfies* (1)-(3) *for the* P_{w_1,w_2,w'_1,w'_2} *specified by* CB_2*, must have* $H(W'_1, W'_2|S) \leq L \log_2(3)$*.*

Proof: Denote the decoding functions of User 1 and User 2 by $\mathcal{F}_{W_1'}$ and $\mathcal{G}_{W_2'}$, respectively. The subscripts indicate that the decoding functions depend on the side-information available to each user. Because we require zero-error decoding, we must have,

$$\mathcal{F}_{W_1'}(S) = W_1, \qquad \mathcal{G}_{W_2'}(S) = W_2 \tag{111}$$

$$\Leftrightarrow \qquad \left[\mathcal{F}_{W_1'}(S)\right]_l = W_1(l), \qquad \left[\mathcal{G}_{W_2'}(S)\right]_l = W_2(l), \qquad \forall l \in [L] \tag{112}$$

where for a length L sequence A, $[A]_l$ denotes the l-th symbol of A.

From (41), we note the following relationship. For any $l \in [L]$,

$$W_1'(l) = 0 \;\; \Rightarrow \;\; W_2(l) = (W_1(l) + W_2'(l)) \mod 4 \tag{113}$$

$$W_1'(l) = 1 \;\; \Rightarrow \;\; W_2(l) = (W_1(l) + 3 - W_2'(l)) \mod 4 \tag{114}$$

We now show that conditioned on any realization of S, and for each index $l \in [L]$, there are only three possible values for the tuple $(W_1'(l), W_2'(l))$. Here is a proof by contradiction. Suppose on the contrary that there exists some realization S^* of S and some index $l^* \in [L]$ such that $(W_1'(l^*), W_2'(l^*))$ can take all 4 values in the set $\{(0,0),(0,1),(1,0),(1,1)\}$. In particular, let A_1, A_2 be the realizations of the length L sequence W_1' and B_1, B_2 be the realizations of the length L sequence W_2' such that $(A_1(l^*), A_2(l^*)) = (0,1)$ and $(B_1(l^*), B_2(l^*)) = (0,1)$. From (112), (113), (114), we have

$$W_1' = A_1, W_2' = B_1 \;\; \Rightarrow \;\; W_1'(l^*) = 0, W_2'(l^*) = 0 \tag{115}$$

$$\overset{(113)}{\Rightarrow} \; W_2(l^*) = (W_1(l^*) + 0) \mod 4 \tag{116}$$

$$\overset{(112)}{\Rightarrow} \; [\mathcal{G}_{B_1}(S^*)]_{l^*} = ([\mathcal{F}_{A_1}(S^*)]_{l^*} + 0) \mod 4 \tag{117}$$

Similarly, $W_1' = A_1, W_2' = B_2 \;\; \Rightarrow \;\; [\mathcal{G}_{B_2}(S^*)]_{l^*} = ([\mathcal{F}_{A_1}(S^*)]_{l^*} + 1) \mod 4$ (118)

$$W_1' = A_2, W_2' = B_1 \;\; \Rightarrow \;\; [\mathcal{G}_{B_1}(S^*)]_{l^*} = ([\mathcal{F}_{A_2}(S^*)]_{l^*} + 3 - 0) \mod 4 \tag{119}$$

$$W_1' = A_2, W_2' = B_2 \;\; \Rightarrow \;\; [\mathcal{G}_{B_2}(S^*)]_{l^*} = ([\mathcal{F}_{A_2}(S^*)]_{l^*} + 3 - 1) \mod 4 \tag{120}$$

Note that (117) - (118) - (119) + (120) gives us $0 = -2 \mod 4$, which is a contradiction. Thus, we have shown that given any realization of S, there are at most 3^L possible realizations of (W_1', W_2'). Using the fact that the uniform distribution maximizes entropy, $H(W_1', W_2' \mid S) \leq L \log_2(3)$ and Lemma 4 is proved.

■

Equipped with Lemma 4, the converse proof is immediate. Let us expand $H(W_1', W_2', S)$ in two ways. On the one hand,

$$\begin{aligned} H(W_1', W_2', S) &= H(W_1', W_2') + H(S|W_1', W_2') && (121)\\ &= 2L + H(S, W_1, W_2|W_1', W_2') && (122)\\ &\geq 2L + H(W_1, W_2|W_1', W_2') && (123)\\ &= 4L && (124) \end{aligned}$$

where (122) follows from the decoding constraints, i.e., from S, W_1', W_2', we can decode W_1, W_2 with no error. On the other hand,

$$H(W_1', W_2', S) = H(S) + H(W_1', W_2'|S) \tag{125}$$

$$\leq \quad H(S) + L\log_2(3) \tag{126}$$

as shown in Lemma 4. Combining (124) and (126), we have

$$\forall L \in \mathbb{N}, \qquad H(S) + L\log_2(3) \geq 4L \tag{127}$$

$$\Rightarrow \qquad \frac{H(S)}{L} \geq 4 - \log_2(3) \tag{128}$$

$$\Rightarrow \qquad H^* = \inf \frac{H(S)}{L} \geq 4 - \log_2(3). \tag{129}$$

and the proof of the converse bound $H^* \geq 4 - \log_2(3)$ is complete.

6.2 Achievability: $H^* \leq 4 - \log_2(3)$ bits

Based on the alphabet, the set of possible values of (w_1', w_2') is $\mathcal{W}_1' \times \mathcal{W}_2' = \{(0,0),(0,1),(1,0),(1,1)\}$. Note that $|\mathcal{W}_1' \times \mathcal{W}_2'| = 4$. Consider an arbitrary sequence of subsets $\mathcal{W}(l) \subset \mathcal{W}_1' \times \mathcal{W}_2'$ such that $|\mathcal{W}(l)| = 3$. First we show that if $(W_1'(l), W_2'(l))$ tuples are restricted to take values in $\mathcal{W}(l)$, then sending $2L$ bits is sufficient to satisfy both users' demands. This result is stated in the following lemma.

Lemma 5 *For any $L \in \mathbb{N}$, if for all $l \in [L]$, the tuple $(W_1'(l), W_2'(l)) \in \mathcal{W}(l) \subset \mathcal{W}_1' \times \mathcal{W}_2'$, $|\mathcal{W}(l)| = 3$, and the sequence $\mathcal{W}(l), l \in [L]$ is already known to the users, then broadcasting $2L$ bits is sufficient to satisfy both users' demands.*

Proof: We have 4 cases for $\mathcal{W}(l)$ as listed below.

1. $\mathcal{W} = \{(0,0),(0,1),(1,0)\}$. In this case, the relationship between $W_2(l)$ and $W_1(l)$ can be described as $W_2(l) = (W_1(l) + 3W_1'(l) + W_2'(l)) \mod 4$ such that transmitting $S(l) = (W_1(l) + 3W_1'(l)) \mod 4$ is sufficient to satisfy both users' demands. User 1 simply subtracts $3W_1'(l)$ (modulo 4) to get $W_1(l)$, and User 2 adds $W_2'(l)$ (modulo 4) to get $W_2(l)$.

2. $\mathcal{W} = \{(0,0),(0,1),(1,1)\}$. Here we have $W_2(l) = (W_1(l) + W_1'(l) + W_2'(l)) \mod 4$ and set $S(l) = (W_1(l) + W_1'(l)) \mod 4$. User 1 simply subtracts $W_1'(l)$ to get $W_1(l)$, and User 2 adds $W_2'(l)$ to get $W_2(l)$, all modulo 4.

3. $\mathcal{W} = \{(0,0),(1,0),(1,1)\}$. Here we have $W_2(l) = (W_1(l) + 3W_1'(l) - W_2'(l)) \mod 4$ and we choose to send $S(l) = (W_1(l) + 3W_1'(l)) \mod 4$. User 1 subtracts $3W_1'(l)$ from $S(l)$ to get $W_1(l)$, and User 2 subtracts $W_2'(l)$ from $S(l)$ to get $W_2(l)$, all modulo 4.

4. $\mathcal{W} = \{(0,1),(1,0),(1,1)\}$. Here we have $W_2(l) = (W_1(l) + W_1'(l) + W_2'(l) + 2) \mod 4$ and we set $S(l) = (W_1(l) + W_1'(l)) \mod 4$. User 1 subtracts $W_1'(l)$ from $S(l)$ to get $W_1(l)$, and User 2 adds $W_2'(l) + 2$ from $S(l)$ to get $W_2(l)$, all modulo 4.

Note that in every case, for each $l \in [L]$, $S(l)$ is a number modulo 4 which is represented by 2 bits, so broadcasting $2L$ bits is sufficient overall. The proof of Lemma 5 is thus complete. ■

The key to the achievable scheme is to send $\mathcal{W}(l)$ to the users, in addition to the $2L$ bits that are needed once $\mathcal{W}(l)$ is known to both users. To describe $\mathcal{W}(l)$ it suffices to describe its complement, i.e., $(\mathcal{W}_1' \times \mathcal{W}_2') \backslash \mathcal{W}(l)$. Equivalently, we wish to describe to the users 1 element of $\mathcal{W}_1' \times \mathcal{W}_2'$ which is not the actual realization of $(W_1'(l), W_2'(l))$ tuple so that the users know that the actual realization

is among the 3 remaining values. Since there are 3 values that are not the actual realization, we have 3 choices for what to send for each $l \in [L]$. Overall, we have 3^L choices for (W_1', W_2') tuples that do not match the actual realization for any $l \in [L]$. We next show that conveying one of these 3^L possibilities (out of the total 4^L possibilities) requires $(2-\log_2(3))L+o(L)$ bits with probability of error $\epsilon \to 0$ as $L \to \infty$. This result is stated in the following lemma with general parameters, which will be used again in the proof of Theorem 5.

Lemma 6 *Suppose there is a set of n_1^L tuples known to a transmitter and receiver, out of which an arbitrary subset of n_2^L tuples are designated acceptable, $n_1, n_2 \in \mathbb{N}, n_2 < n_1$. The acceptable tuples are known only to the transmitter, and the goal is for the transmitter to communicate any one of these acceptable tuples to the receiver. Then there exists an ϵ-error scheme that allows the transmitter to accomplish this task by sending only $(\log_2(n_1)-\log_2(n_2))L+o(L)$ bits to the receiver.*

The detailed proof of Lemma 6 is deferred to Section 6.3. Let us present an outline of the proof here. The scheme is based on random binning. Throw the n_1^L tuples uniformly into roughly n_2^L bins. Pick bin 1. Find an acceptable tuple in bin 1 and send its index. Because there are n_2^L bins and n_2^L acceptable tuples, an ϵ change in the exponents will guarantee that each bin will typically get at least one acceptable tuple with high probability. Specifying the index of the acceptable tuple will take $\log_2(n_1^L/n_2^L) = (\log_2(n_1)-\log_2(n_2))L$ bits because each bin contains approximately n_1^L/n_2^L tuples.

Finally, let us summarize the overall achievable scheme which requires a minor adjustment to make it a zero-error scheme. For each realization of (W_1, W_2, W_1', W_2'), we use the scheme from Lemma 6 to find and specify one acceptable (W_1', W_2') tuple, i.e., a tuple that does not match the actual realization of $(W_1'(l), W_2'(l)$ for any $l \in [L]$ to both users. With probability $1-\epsilon$, an acceptable (W_1', W_2') tuple is found and specified, and then we use the scheme from Lemma 5 so that each user decodes the desired message. The total number of bits broadcast in this case is $(2-\log_2(3))L + o(L) + 2L$. With probability ϵ, we do not find an acceptable (W_1', W_2') tuple. In this case, we directly send (W_1, W_2), and the number of bits broadcast is $8L$ bits. Therefore, the average number of bits broadcast to the users is

$$(1-\epsilon) \times [(4-\log_2(3))L + o(L)] + \epsilon \times 8L + 1 \tag{130}$$

where 1 extra bit is used to specify if an acceptable (W_1', W_2') tuple is found. This implies that

$$H(S) \leq (1-\epsilon) \times [(4-\log_2(3))L + o(L)] + \epsilon \times 8L + 1 \tag{131}$$

$$\Rightarrow H^* = \inf \frac{H(S)}{L} \leq 4 - \log_2(3). \tag{132}$$

The achievability proof, i.e., the proof of the bound $H^* \leq 4-\log_2(3)$ bits, is thus complete. ■

Combining the converse and achievability proofs we have shown that $H^* = 4-\log_2(3)$ bits, which implies that $C_{CB_2} = \frac{H(w_1,w_2)}{H^*} = \frac{4}{4-\log_2(3)}$ by definition.

6.3 Proof of Lemma 6

Fix $L \in \mathbb{N}$ and $\delta = \frac{1}{\sqrt{L}}$ such that $L(1-\delta)$ is an integer. We have n_1^L tuples and $n_2^{L(1-\delta)}$ bins. For each tuple, choose a bin index independently and uniformly over $[n_2^{L(1-\delta)}]$. Denote the bin index of the i-th tuple by $X_i, i \in [n_1^L]$, so X_i is uniformly distributed over $[n_2^{L(1-\delta)}]$.

The number of tuples with bin index 1 is $T_1 = \sum_{i\in[n_1^L]} \mathbb{1}(X_i = 1)$. Its expected value and variance are computed as follows.

$$\mu_1 = \mathbb{E}\left[\sum_{i\in[n_1^L]} \mathbb{1}(X_i = 1)\right] = \sum_{i\in[n_1^L]} \mathbb{E}\left[\mathbb{1}(X_i = 1)\right] = \frac{n_1^L}{n_2^{L(1-\delta)}} \tag{133}$$

$$\begin{aligned}\sigma_1^2 &= \mathbb{E}\left[\left(\sum_{i\in[n_1^L]} \mathbb{1}(X_i = 1)\right)^2\right] - \mu_1^2 = \mathbb{E}\left[\left(\sum_{i\in[n_1^L]} \mathbb{1}(X_i = 1)\right)\left(\sum_{j\in[n_1^L]} \mathbb{1}(X_j = 1)\right)\right] - \mu_1^2 \\ &= \sum_{i\in[n_1^L]} \mathbb{E}\left[\left(\mathbb{1}(X_i = 1)\right)^2\right] + \sum_{i\neq j, i,j\in[n_1^L]} \mathbb{E}\left[\mathbb{1}(X_i = 1)(X_j = 1)\right] - \mu_1^2 \qquad (134) \\ &= \frac{n_1^L}{n_2^{L(1-\delta)}} + \frac{n_1^{2L} - n_1^L}{n_2^{2L(1-\delta)}} - \frac{n_1^{2L}}{n_2^{2L(1-\delta)}} = n_1^L\left(\frac{1}{n_2^{L(1-\delta)}} - \frac{1}{n_2^{2L(1-\delta)}}\right) \qquad (135)\end{aligned}$$

From Chebyshev's inequality, we have

$$\Pr(T_1 \geq (1+\delta)\mu_1) \leq \frac{\sigma_1^2}{\delta^2\mu_1^2} = \frac{n_1^L\left(\frac{1}{n_2^{L(1-\delta)}} - \frac{1}{n_2^{2L(1-\delta)}}\right)}{\delta^2 \frac{n_1^{2L}}{n_2^{2L(1-\delta)}}} = \frac{n_2^{L(1-\delta)} - 1}{\delta^2 n_1^L} \tag{136}$$

Therefore, for any small constant ϵ, we can find a sufficiently large L such that

$$\Pr(T_1 \geq (1+\delta)\mu_1) \leq \epsilon/2 \tag{137}$$

Consider any n_2^L acceptable tuples. Denote the bin index for the i-th acceptable tuple by $Y_i, i \in [n_2^L]$, and Y_i is also uniform over $[n_2^{L(1-\delta)}]$. We similarly consider the number of acceptable tuples with bin index 1, denoted as $T_2 = \sum_{i\in[n_2^L]} \mathbb{1}(Y_i = 1)$.

$$\mu_2 = \mathbb{E}\left[T_2\right] = n_2^{L\delta}, \sigma_2^2 = \mathbb{E}\left[T_2^2\right] - \mu_2^2 = n_2^{L\delta}(1 - n_2^{-L(1-\delta)}) \tag{138}$$

$$\Pr(T_2 = 0) \leq \Pr(|T_2 - \mu_2| \geq \delta\mu_2) \leq \frac{1 - n_2^{-L(1-\delta)}}{\delta^2 n_2^{L\delta}} \leq \epsilon/2 \tag{139}$$

The coding scheme works as follows. When the number of tuples in bin 1, i.e., $T_1 \geq (1+\delta)\mu_1$, declare an error. If there is no acceptable tuple in bin 1 ($T_2 = 0$), declare an error. Otherwise, we send the index of any acceptable tuple. From (137), (139) and the union bound, the error probability is no larger than $\epsilon/2 + \epsilon/2 = \epsilon$, which can be made arbitrarily small by picking a sufficiently large L.

Finally, we compute the number of bits used. Note that $\delta = 1/\sqrt{L}$.

$$\begin{aligned}\log_2\left((1+\delta)\mu_1\right) &= \log_2(1+\delta) + \log_2\left(\frac{n_1^L}{n_2^L n_2^{-L\delta}}\right) = L\log_2\left(\frac{n_1}{n_2}\right) + \sqrt{L}\log_2(n_2) + \log_2(1 + \frac{1}{\sqrt{L}}) \\ &= L(\log_2(n_1) - \log_2(n_2)) + o(L) \qquad (140)\end{aligned}$$

Therefore, the number of bits used matches that in the lemma. The proof of Lemma 6 is complete. ■

7 Proof of Theorem 5

For the proof, it will be less cumbersome to work with the optimal normalized broadcast cost as defined in (110). Specifically, we first prove that $\log_2(m)$ bits $\leq H^* \leq \log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)$ bits in Section 7.1. Then we show that the upper extreme is tight for minimally structured settings in Section 7.2, and that the lower extreme is tight if the setting is maximally structured in Section 7.3.

7.1 $\log_2(m)$ bits $\leq H^* \leq \log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)$ bits

The lower bound, $H^* \geq \log_2(m)$ bits follows immediately from Theorem 2. The bound is quite obvious, as $H^* \geq H(w_1|w_1') = \log_2(m)$. The remainder of this section is aimed at proving the upper bound, $H^* \leq \log_2(m)+\log_2(m_1 m_2)-\log_2(m_1+m_2-1)$ bits. We will construct an achievable scheme that works for all settings of matching computation broadcast. To this end, let us introduce some definitions along with illustrative examples. Without loss of generality, we will assume $m_1 \geq m_2$.

Definition 6 (Standard Form, •-Set and ∘-Set) *Let us attach a label to each element $(a_i, b_j), i \in [m_1], j \in [m_2]$ of the Π matrix as follows. The (a_i, b_j) element is labelled with $\bullet$ if $b_j = 1$ or if $b_j = a_i + 1$. Otherwise, label it with $\circ$. We will refer to this labelling of Π as the standard form. The set of (a_i, b_j) with label $\bullet$ is called the $\bullet$-set and the set of (a_i, b_j) with label $\circ$ is called the $\circ$-set. Note that the cardinality of the $\bullet$-set is $m_1 + m_2 - 1$ and the $\circ$-set is the complement of the $\bullet$-set.*

For example, when $m_1 = 3, m_2 = 2$, the standard form of Π, •-set and ∘-set are shown below.

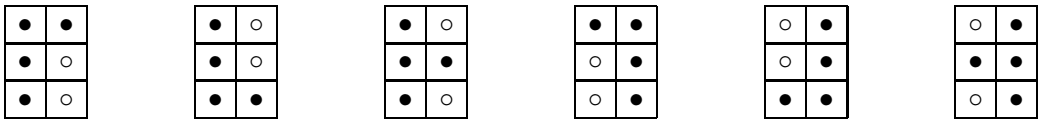
	$w_2' = 1$	$w_2' = 2$
$w_1' = 1$	$\bullet\ \pi_{1,1}$	$\bullet\ \pi_{1,2}$
$w_1' = 2$	$\bullet\ \pi_{2,1}$	$\circ\ \pi_{2,2}$
$w_1' = 3$	$\bullet\ \pi_{3,1}$	$\circ\ \pi_{3,2}$

(standard form)

$\bullet - \text{set} : \{(1,1),(2,1),(3,1),(1,2)\}$
$\circ - \text{set} : \{(2,2),(3,2)\}$

Definition 7 (Translation) *Consider any cyclic shift of the rows and/or columns of Π labelled in standard form, i.e., $\forall i \in [m_1]$, the i-th row is shifted to the $((i + z_1) \mod m_1)$-th row and $\forall j \in [m_2]$, the j-th column is shifted to the $((j + z_2) \mod m_2)$-th column, $i, z_1 \in [m_1], j, z_2 \in [m_2]$. The resulting $\bullet$-set and $\circ$-set are called* translations.

For example, when $m_1 = 3, m_2 = 2$, all possible translations of the •-set and the ∘-set are shown below.



where the first translation is the original standard form, and the second translation is obtained by setting $z_1 = 2, z_2 = 2 = 0 \mod 2$ (rows are cyclicly shifted by 2 and columns are not shifted).

Following the notion in geometry, translation refers to a function that moves an object without rotating or flipping it. Intuitively, we may think of it as replicating the standard form grid pattern infinitely in space, and choosing any contiguous $m_1 \times m_2$ block from that infinite grid. Such a block is a translation.

For our achievable scheme, we will only consider the •-sets and ∘-sets that can be obtained by translations of the standard form. Such •-sets and ∘-sets are called regular •-sets and regular

$\circ$-sets, respectively. The importance of regular $\bullet$-sets is highlighted in the following lemma, where we show that if (w_1', w_2') can only take values from a regular $\bullet$-set, then sending $\log_2(m)$ bits per symbol is sufficient to satisfy both users' demands. Essentially, the following lemma generalizes Lemma 5.

Lemma 7 *For any $L \in \mathbb{N}$, if for all $l \in [L]$, the tuple $(W_1'(l), W_2'(l)) \in \mathcal{W}(l) \subset [m_1] \times [m_2]$, each $\mathcal{W}(l)$ is a regular $\bullet$-set, and the sequence $\mathcal{W}(l), l \in [L]$ is already known to the users, then broadcasting $L \log_2(m)$ bits is sufficient to satisfy both users' demands.*

Proof: For any L, consider an arbitrary regular $\bullet$-set with cyclic shifts z_1, z_2 so that the $\bullet$-set contains the following elements.

$$\begin{aligned} \bullet\text{-set} \quad = \quad & \{(1+z_1, 1+z_2), (2+z_1, 1+z_2), \cdots, (m_1+z_1, 1+z_2), \\ & (1+z_1, 2+z_2), (2+z_1, 3+z_2), \cdots, (m_2-1+z_1, m_2+z_2)\} \end{aligned} \tag{141}$$

where for an element $(a_i, b_j) \in \bullet$-set, a_i is interpreted modulo m_1 and b_j is interpreted modulo m_2.

We show that there exist $m_1 + m_2$ permutations $\delta_1, \cdots, \delta_{m_1}, \gamma_1, \cdots, \gamma_{m_2}$ over $[m]$ such that the following equation holds.

$$\gamma_{w_2'} \delta_{w_1'} = \pi_{w_1', w_2'}, \forall (w_1', w_2') \in \bullet\text{-set} \tag{142}$$

Such $\delta_i, \gamma_j, i \in [m_1], j \in [m_2]$ are chosen as follows.

$$\begin{aligned} &\text{Choose } \gamma_{1+z_2} \text{ to be an arbitrary permutation, say identity.} \\ &\text{Set } \delta_{1+z_1} = \gamma_{1+z_2}^{-1} \pi_{1+z_1, 1+z_2} \text{ such that } \gamma_{1+z_2} \delta_{1+z_1} = \pi_{1+z_1, 1+z_2} \\ &\text{Set } \delta_{2+z_1} = \gamma_{1+z_2}^{-1} \pi_{2+z_1, 1+z_2} \text{ such that } \gamma_{1+z_2} \delta_{2+z_1} = \pi_{2+z_1, 1+z_2} \\ &\qquad\qquad \vdots \\ &\text{Set } \delta_{m_1+z_1} = \gamma_{1+z_2}^{-1} \pi_{m_1+z_1, 1+z_2} \text{ such that } \gamma_{1+z_2} \delta_{m_1+z_1} = \pi_{m_1+z_1, 1+z_2} \\ &\text{Set } \gamma_{2+z_2} = \pi_{1+z_1, 2+z_2} \delta_{1+z_1}^{-1} \text{ such that } \gamma_{2+z_2} \delta_{1+z_1} = \pi_{1+z_1, 2+z_2} \\ &\text{Set } \gamma_{3+z_2} = \pi_{2+z_1, 3+z_2} \delta_{2+z_1}^{-1} \text{ such that } \gamma_{3+z_2} \delta_{2+z_1} = \pi_{2+z_1, 3+z_2} \\ &\qquad\qquad \vdots \\ &\text{Set } \gamma_{m_2+z_2} = \pi_{m_2-1+z_1, m_2+z_2} \delta_{m_2-1+z_1}^{-1} \text{ such that } \gamma_{m_2+z_2} \delta_{m_2-1+z_1} = \pi_{m_2-1+z_1, m_2+z_2} \end{aligned} \tag{143}$$

where we interpret the index of δ_i modulo m_1 and the index of γ_j modulo m_2. It is easy to verify that with the choice of δ_i, γ_j in (143), (142) is satisfied. The choices of δ_i, γ_j for any regular $\bullet$-set are fixed and known globally. The achievable scheme now works as follows.

For any realization of $(W_1(l), W_2(l), W_1'(l), W_2'(l))$, we send $S(l) = \delta_{W_1'(l)}(W_1(l))$, which contains $\log_2(m)$ bits. Both users decode their desired messages using the following structured decoding rule. User 1 takes the received $\delta_{W_1'(l)}(W_1(l))$ and applies the permutation $\delta_{W_1'(l)}^{-1}$ to obtain $W_1(l)$. User 2 takes the received $\delta_{W_1'(l)}(W_1(l))$ and applies the permutation $\gamma_{W_2'(l)}$ to obtain

$$\gamma_{W_2'(l)} \delta_{W_1'(l)}(W_1(l)) \overset{(142)}{=} \pi_{W_1'(l), W_2'(l)}(W_1(l)) \overset{(43)}{=} W_2(l) \tag{144}$$

Note that $(W_1'(l), W_2'(l)) \in \bullet$-set. Repeating the scheme above for all $l \in [L]$ gives us the zero-error scheme that broadcasts $L \log_2(m)$ bits. This completes the proof of Lemma 7. ■

To complete the description of the general achievable scheme we must also send some information so that for each $l \in [L]$, the users know *one* regular $\bullet$-set that includes the actual realization of $(W_1'(l), W_2'(l))$, so that we can apply the scheme in Lemma 7. Such a regular $\bullet$-set is called *acceptable*. For example, suppose $m_1 = 3, m_2 = 2$ and the actual realization of $(W_1'(1), W_2'(1))$ is $(2, 1)$. Then the acceptable regular $\bullet$-set must contain $(2, 1)$, which is indicated with a shaded gray region below. So the only acceptable $\bullet$-sets are the following $4(= m_1 + m_2 - 1)$.

In general, for any realization of $(W_1'(l), W_2'(l))$, let us show that there are $(m_1 + m_2 - 1)$ acceptable regular $\bullet$-sets. This result is stated in the following lemma.

Lemma 8 *For any L and any realization of (W_1', W_2'), there are $(m_1 + m_2 - 1)^L$ acceptable regular $\bullet$-sets, out of all $(m_1 m_2)^L$ regular $\bullet$-sets.*

Proof: We first show that for any $l \in [L]$, there are $m_1 m_2$ regular $\bullet$-sets. To this end, it suffices to show that all translations of the standard form produce distinct regular $\bullet$-sets. Consider two translated $\bullet$-sets with cyclic shifts, $(z_1, z_2), (z_1', z_2')$ such that $z_1, z_1' \in [m_1], z_2, z_2' \in [m_2], (z_1, z_2) \neq (z_1', z_2')$. Note that the $\bullet$-set in standard form contains a column where each element is labelled by $\bullet$, so if $z_2 \neq z_2'$, the two translated $\bullet$-sets are distinct (the column with all $\bullet$ is different). Now consider the case where $z_2 = z_2'$ while $z_1 \neq z_1'$. Here the two translated $\bullet$-sets are again distinct because the first row of the $\bullet$-set in standard form is distinct from all other rows and as $z_1 \neq z_1'$, the first row is shifted to distinct rows. Thus in total, we have $(m_1 m_2)^L$ regular sets.

Next we show that for any $l \in [L]$ and any realization $(i^*, j^*) \in [m_1] \times [m_2]$ of $(W_1'(l), W_2'(l))$, there are $m_1 + m_2 - 1$ acceptable regular $\bullet$-sets. To see this, note that there are $m_1 + m_2 - 1$ distinct elements labelled with a $\bullet$ in the standard form. We may shift each element to (i^*, j^*), and each such shift corresponds to a distinct translation. Thus in total, we have $(m_1 + m_2 - 1)^L$ acceptable regular $\bullet$-sets. This completes the proof of Lemma 8. ■

Combining Lemma 8 and Lemma 6, we know that communicating an acceptable regular $\bullet$-set to the users requires $L(\log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1))$ bits with probability $1 - \epsilon$. The overall achievable scheme is described as follows. For each realization of (W_1, W_2, W_1', W_2'), we use the scheme from Lemma 6 to find and specify one acceptable regular $\bullet$-set. With probability $1 - \epsilon$, an acceptable regular $\bullet$-set is found, and then we use the scheme from Lemma 7 so that each user decodes the desired message. The number of bits broadcast is $L(1 - \epsilon)[\log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1) + \log_2(m)]$. For the remaining probability ϵ, we directly send (W_1, W_2). The number of bits broadcast is $2L\epsilon \log_2(m)$ bits. One extra bit is used to identify the cases where (W_1, W_2) are directly sent. Therefore,

$$H^* = \inf_{L \in \mathbb{N}} \frac{H(S)}{L} \leq \frac{(1 - \epsilon) \times [(\log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1))]L + \epsilon \times 2L \log_2(m) + 1}{L} \tag{145}$$

and since $\epsilon \to 0$ as $L \to \infty$, we have

$$H^* \leq \log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1). \tag{146}$$

The achievable scheme requires $L \to \infty$ (mainly because of the random binning operation in Lemma 6) which is sufficient for our purpose. However, non-asymptotic schemes may also be possible. To show this, let us show an example of a finite L scheme when $m_1 = 4, m_2 = 3$ ($L = 1$ in fact). This example is special because $\log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)$ takes an integer value of 1.

A non-asymptotic scheme when $m_1 = 4, m_2 = 3$

While this scheme uses similar ideas as the asymptotic scheme, it is based on a different definition of the $\bullet$-set (not obtained by translations from standard form). Specifically for this example, the $\bullet$-set is defined as follows.

	γ_1	γ_2	γ_3
δ_1	$\bullet$ $\pi_{1,1}$	$\circ$ $\pi_{1,2}$	$\circ$ $\pi_{1,3}$
δ_2	$\bullet$ $\pi_{2,1}$	$\bullet$ $\pi_{2,2}$	$\circ$ $\pi_{2,3}$
δ_3	$\circ$ $\pi_{3,1}$	$\bullet$ $\pi_{3,2}$	$\bullet$ $\pi_{3,3}$
δ_4	$\circ$ $\pi_{4,1}$	$\circ$ $\pi_{4,2}$	$\bullet$ $\pi_{4,3}$

Note that we label the rows and columns by the permutations δ_i, γ_j that we assign as follows to satisfy $\gamma_j \delta_i = \pi_{i,j}$ if $(i,j) \in \bullet$-set (following the same idea from Lemma 7).

$$\begin{aligned} &\text{Choose } \gamma_1 \text{ to be an arbitrary permutation} \\ &\text{Set } \delta_1 = \gamma_1^{-1}\pi_{1,1}, \delta_2 = \gamma_1^{-1}\pi_{2,1}, \gamma_2 = \pi_{2,2}\delta_2^{-1} \\ &\text{Set } \delta_3 = \gamma_2^{-1}\pi_{3,2}, \gamma_3 = \pi_{3,3}\delta_3^{-1}, \delta_4 = \gamma_3^{-1}\pi_{4,3} \end{aligned} \tag{147}$$

If the users know that $(W_1'(1), W_2'(1)) \in \bullet$-set, then sending $\delta_{W_1'(1)}(W_1(1))$ ($= \log_2(m)$ bits) is sufficient to satisfy both users's demands. After receiving $\delta_{W_1'(1)}(W_1(1))$, User 1 applies $\delta_{W_1'(1)}^{-1}$ to obtain $W_1(1)$, and User 2 applies $\gamma_{W_2'(1)}$ to obtain $\gamma_{W_2'(1)}\delta_{W_1'(1)}(W_1(1)) = \pi_{W_1'(1),W_2'(1)}(W_1(1)) = W_2(1)$. Interestingly, if $(W_1'(1), W_2'(1)) \in \circ$-set, we may assign δ_i, γ_j (differently) as follows such that $\gamma_j \delta_i = \pi_{i,j}$ if $(i,j) \in \circ$-set and sending $\delta_{W_1'(1)}(W_1(1))$ is sufficient to satisfy both users' demands.

$$\begin{aligned} &\text{Choose } \gamma_1 \text{ to be an arbitrary permutation} \\ &\text{Set } \delta_3 = \gamma_1^{-1}\pi_{3,1}, \delta_4 = \gamma_1^{-1}\pi_{4,1}, \gamma_2 = \pi_{4,2}\delta_4^{-1} \\ &\text{Set } \delta_1 = \gamma_2^{-1}\pi_{1,2}, \gamma_3 = \pi_{1,3}\delta_1^{-1}, \delta_2 = \gamma_3^{-1}\pi_{2,3} \end{aligned} \tag{148}$$

The only remaining step is to send information so that the users know $(W_1'(1), W_2'(1))$ belong to $\bullet$-set or $\circ$-set, for which 1 bit is sufficient. The broadcast cost thus achieved is $\log_2(m) + 1 = \log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)$ bits, which matches the optimal value H^*.

7.2 $H^* = \log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)$ bits if Minimally Structured

We show that for minimally structured settings, the general achievable scheme described in Section 7.1 is the best possible, i.e., $H^* \geq \log_2(m) + \log_2(m_1 m_2) - \log_2(m_1 + m_2 - 1)$ bits.

We start with a lemma, which is a generalization of Lemma 4. Interpreted through the lens of induced permutations, Lemma 4 states that if the induced permutation of a length-4 cycle is a derangement, then given S the set of feasible (W_1', W_2') tuple values can not include all the terms of the cycle. The following lemma generalizes the same argument to cycles of any length. For simplicity, if the induced permutation of a cycle is a derangement, we say that the cycle is a derangement cycle.

Lemma 9 *For any given realization of S and for any symbol index $l \in [L]$, the set of feasible values for $(W_1'(l), W_2'(l))$ contains no derangement cycle.*

Proof: The proof is by contradiction. So, let us assume that for some given realization S^* of S, and some $l^* \in [L]$, the set of feasible values of $(W_1'(l^*), W_2'(l^*))$ contains a cycle of length N,

$$(a_1, b_1) \leftrightarrow (a_2, b_2) \leftrightarrow \cdots \leftrightarrow (a_N, b_N) \leftrightarrow (a_1, b_1) \tag{149}$$

Thus, the feasible values for $(W_1'(l^*), W_2'(l^*))$ include all of the values in the set $\{(a_1, b_1), (a_2, b_2), \cdots, (a_N, b_N)\}$. Let $A_1, A_2, \cdots, A_N$ denote the corresponding realizations of W_1', so that we have $A_j(l^*) = a_j, j \in [N]$, and $B_1, B_2, \cdots, B_N$ denote the corresponding realizations of W_2' such that $B_j(l^*) = b_j$. If $a_j = a_k$ then $A_j = A_k$, and if $b_j = b_k$ then $B_j = B_k$. Recall that $\mathcal{F}, \mathcal{G}$ denote the decoding functions of users 1 and 2, respectively. Based on the structure of the matching computation broadcast problem (43) and the zero-error decoding constraint (1), (2), we have

$$\begin{aligned}
[\mathcal{G}_{B_1}(S^*)]_{l^*} &= \pi_{a_1,b_1} [\mathcal{F}_{A_1}(S^*)]_{l^*} \\
[\mathcal{G}_{B_2}(S^*)]_{l^*} &= \pi_{a_2,b_2} [\mathcal{F}_{A_2}(S^*)]_{l^*} \\
[\mathcal{G}_{B_3}(S^*)]_{l^*} &= \pi_{a_3,b_3} [\mathcal{F}_{A_3}(S^*)]_{l^*} \\
&\vdots \\
[\mathcal{G}_{B_N}(S^*)]_{l^*} &= \pi_{a_N,b_N} [\mathcal{F}_{A_N}(S^*)]_{l^*}
\end{aligned} \tag{150}$$

From the definition of a cycle, it follows that

$$\begin{aligned}
a_1 = a_2 &\Rightarrow A_1 = A_2 \\
b_2 = b_3 &\Rightarrow B_2 = B_3 \\
a_3 = a_4 &\Rightarrow A_3 = A_4 \\
b_4 = b_5 &\Rightarrow B_4 = B_5 \\
\vdots \quad &\vdots \\
a_{N-1} = a_N &\Rightarrow A_{N-1} = A_N \\
b_N = b_1 &\Rightarrow B_N = B_1
\end{aligned} \tag{151}$$

Combining (150) and (151), we have

$$\begin{aligned}
[\mathcal{G}_{B_N}(S^*)]_{l^*} &= \pi_{a_1,b_N} [\mathcal{F}_{A_1}(S^*)]_{l^*} \\
[\mathcal{G}_{B_2}(S^*)]_{l^*} &= \pi_{a_1,b_2} [\mathcal{F}_{A_1}(S^*)]_{l^*} \\
[\mathcal{G}_{B_2}(S^*)]_{l^*} &= \pi_{a_3,b_2} [\mathcal{F}_{A_3}(S^*)]_{l^*} \\
&\vdots \\
[\mathcal{G}_{B_N}(\vec{s})]_{i^*} &= \pi_{a_{N-1},b_N} \left[\mathcal{F}_{A_{N-1}}(S^*)\right]_{l^*}
\end{aligned} \tag{152}$$

which implies that

$$[\mathcal{G}_{B_N}(S^*)]_{l^*} = \pi_{a_1,b_N} \pi_{a_1,b_2}^{-1} \pi_{a_3,b_2} \cdots \pi_{a_{N-1}b_N}^{-1} [\mathcal{G}_{B_N}(S^*)]_{l^*} \tag{153}$$

Note that the cycle is a derangement cycle, so the induced permutation $\pi_{a_1,b_N} \pi_{a_1,b_2}^{-1} \pi_{a_3,b_2} \cdots \pi_{a_{N-1}b_N}^{-1}$ is a derangement, i.e., there is no fixed point.

However, note that

$$[\mathcal{G}_{B_N}(S^*)]_{l^*} = W_2(l^*) = \pi_{a_1,b_N}\pi^{-1}_{a_1,b_2}\pi_{a_3,b_2}\cdots\pi^{-1}_{a_{N-1}b_N}(W_2(l^*)),$$

so the decoding is incorrect. Thus, we arrive at the contradiction which completes the proof of Lemma 9. ■

Note that Lemma 9 holds in general, e.g., it is not limited to minimally structured settings. Next, for minimally structured settings we show that if a set of values for $(W_1'(l), W_2'(l))$ contains no derangement cycle, then the cardinality of the set is no more than $m_1 + m_2 - 1$. The intuitive reason is that a set of values for $(W_1'(l), W_2'(l))$ with more than m_1+m_2-1 elements over $[m_1]\times[m_2]$ must contain a cycle and every cycle is a derangement cycle for minimally structured settings. This result is stated in the following lemma.

Lemma 10 *For minimally structured settings, if the set $\mathcal{M} \subset [m_1]\times[m_2]$ contains no derangement cycle, then*

$$|\mathcal{M}| \leq m_1 + m_2 - 1 \tag{154}$$

Proof: Since every cycle for a minimally structured setting is a derangement cycle, we only need to show that $|\mathcal{M}| \leq m_1+m_2-1$ for cycle-free $\mathcal{M} \subset [m_1]\times[m_2]$. Let the elements of $[m_1]\times[m_2]$ be mapped to the $m_1 \times m_2$ table under the natural ordering. Remove any rows or columns of the table that have no elements of $\mathcal{M}$, leaving us with $m_1' \leq m_1$ rows and $m_2' \leq m_2$ columns. This cannot introduce cycles, so it suffices to show that $|\mathcal{M}| \leq m_1'+m_2'-1$, for cycle-free $\mathcal{M} \subset [m_1']\times[m_2']$. This is equivalent to the original statement of the lemma, so without loss of generality we can assume that $(m_1', m_2') = (m_1, m_2)$. Now, find a row or a column of the table that has exactly 1 element of $\mathcal{M}$. There must exist such a row or column, because otherwise $\mathcal{M}$ contains a cycle. Eliminate this row or column, and remove the corresponding element from $\mathcal{M}$. So it now remains to show that $|\mathcal{M}| - 1 \leq m_1 + m_2 - 2$, which is also equivalent to the original statement, i.e., the proof for the reduced setting implies the proof for the original setting. Continue this step, until there remains only one row or only one column. Without loss of generality, suppose in the end we have m_1 rows and one column. Then we only have to show that any subset of this table cannot have more than m_1 elements, which is trivially true. Hence, Lemma 10 is proved. ■

The converse proof is a simple consequence of the above two lemmas. From Lemma 9 and Lemma 10, we know that given any realization of S, the number of feasible values for (W_1', W_2') is no more than $(m_1 + m_2 - 1)^L$, i.e., $H(W_1', W_2'|S) \leq L\log_2(m_1 + m_2 - 1)$. Then we expand $H(S, W_1', W_2)$ in two ways, similar to the proof of Lemma 1.

$$\begin{aligned} H(S, W_1', W_2) &= H(W_1', W_2') + H(S|W_1', W_2') = L\log_2(m_1m_2) + L\log_2(m) && (155)\\ &= H(S) + H(W_1', W_2'|S) \leq H(S) + L\log_2(m_1 + m_2 - 1) && (156)\\ \Rightarrow H(S)/L &\geq \log_2(m) + \log_2(m_1m_2) - \log_2(m_1 + m_2 - 1) && (157)\end{aligned}$$

As $H^* = \inf H(S)/L$, the desired bound follows and the proof of the converse bound, $H^* \geq \log_2(m) + \log_2(m_1m_2) - \log_2(m_1 + m_2 - 1)$ bits for minimally structured settings is thus complete.

7.3 $H^* = \log_2(m)$ bits if Maximally Structured

We show that for maximally structured settings, the broadcast cost $\log_2(m)$ is achievable, which is a simple consequence of Lemma 7. Specifically, we show that although the choice of δ_i, σ_j in

Lemma 7 (refer to (143)) is designed to satisfy

$$\gamma_{w_2'}\delta_{w_1'} = \pi_{w_1',w_2'} \tag{158}$$

for all (w_1', w_2') from only a $\bullet$-set (refer to (142)), in fact it automatically satisfies (158) for all $(w_1', w_2') \in [m_1] \times [m_2]$ if the setting is maximally structured. Specifically, following (143), we proceed as follows.

$$\begin{array}{l} \text{Choose } \gamma_1 \text{ to be an arbitrary permutation} \\ \text{Set } \delta_1 = \gamma_1^{-1}\pi_{1,1}, \delta_2 = \gamma_1^{-1}\pi_{2,1}, \cdots, \delta_{m_1} = \gamma_1^{-1}\pi_{m_1,1} \\ \text{Set } \gamma_2 = \pi_{1,2}\delta_1^{-1}, \gamma_3 = \pi_{2,3}\delta_2^{-1}, \cdots, \gamma_{m_2} = \pi_{m_2-1,m_2}\delta_{m_2-1}^{-1} \end{array} \tag{159}$$

and show that (158) is satisfied for all $(w_1', w_2') \in [m_1] \times [m_2]$ for maximally structured settings. For any $(w_1', w_2') \in [m_1] \times [m_2]$, we have a length-4 cycle $(w_1'-1, 1) \leftrightarrow (w_1'-1, w_2') \leftrightarrow (w_1', w_2') \leftrightarrow (w_1', 1) \leftrightarrow (w_1'-1, 1)$. As the setting is maximally structured, the induced permutation $\pi_{w_1'-1,1}\pi^{-1}_{w_1'-1,w_2'}\pi_{w_1',w_2'}\pi^{-1}_{w_1',1}$ is an identity. We have

$$\begin{aligned} \text{Identity} &= \pi_{w_1'-1,1}\pi^{-1}_{w_1'-1,w_2'}\pi_{w_1',w_2'}\pi^{-1}_{w_1',1} && (160)\\ &\overset{(159)}{=} \gamma_1\delta_{w_1'-1}(\gamma_{w_2'}\delta_{w_1'-1})^{-1}\pi_{w_1',w_2'}(\gamma_1\delta_{w_1'})^{-1} && (161)\\ &= \gamma_1\delta_{w_1'-1}\delta^{-1}_{w_1'-1}\gamma^{-1}_{w_2'}\pi_{w_1',w_2'}\delta^{-1}_{w_1'}\gamma_1^{-1} && (162)\\ \Rightarrow \gamma_{w_2'}\delta_{w_1'} &= \pi_{w_1',w_2'} && (163) \end{aligned}$$

so that (158) is satisfied for all $(w_1', w_2') \in [m_1] \times [m_2]$.

The remaining description of the achievable scheme is the same as that in Lemma 7. For any $l \in [L]$, we send $S(l) = \delta_{W_1'(l)}(W_1(l))$, which requires $\log_2(m)$ bits. User 1 takes the received $\delta_{W_1'(l)}(W_1(l))$ and applies the permutation $\delta^{-1}_{W_1'(l)}$ to obtain $W_1(l)$. User 2 takes the received $\delta_{W_1'(l)}(W_1(l))$ and applies the permutation $\gamma_{W_2'(l)}$ to obtain

$$\gamma_{W_2'(l)}\delta_{W_1'(l)}(W_1(l)) \overset{(158)}{=} \pi_{W_1'(l),W_2'(l)}(W_1(l)) \overset{(43)}{=} W_2(l). \tag{164}$$

The broadcast cost thus achieved is $\log_2(m)$ bits. For maximally structured settings, we note that it suffices to set $L = 1$ because there is no need to send additional information in the manner of Lemma 8. The proof that $H^* = \log_2(m)$ bits for maximally structured settings, is thus complete.

8 Conclusion

The computation broadcast problem represents a small step towards an understanding of the dependencies that exist across message flows and side-informations when communication networks are used for distributed computing applications. Since linear computations are quite common, the capacity characterization for the linear computation broadcast problem is significant. The immediate question for future work is to find the capacity of linear computation broadcast for more than 2 users. The question is particularly interesting because even the 3 user setting appears to be non-trivial, i.e., it does not follow as a direct extension from the 2 user case studied here. Beyond linear settings, a number of questions remain open even for 2 users. While the general converse

bound of Theorem 2 uses only entropic structure, it is not known if it captures *all* of the entropic structure, i.e., whether the bound is always tight for the entropic formulation of the computation broadcast problem. Another interesting problem is to use the insights from the linear and matching computation broadcast problems to construct powerful achievable schemes for general computation broadcast, even for two users. For example, is it possible to create an efficient a, b, c partition of a general computation broadcast problem? If so, then the optimal solutions for a and c partitions are already known in the general case, which leaves us with only the b partition, i.e., the minimally dependent part of the problem. The matching problems appear to be the key to the general solution of such settings. The exact capacity for matching computation broadcast problems also remains open for settings that are neither maximally structured nor minimally structured. A remarkable insight from the capacity characterization for minimally structured settings is that it is better to exploit local structure even with the additional overhead cost of identifying this local structure to both receivers (this overhead is the greatest in minimally structured settings), rather than the obvious alternative, which is to ignore the minimal structure and simply use random coding. The possibility of generalizing this intuition to broader classes of computation broadcast is worth exploring as well. Evidently, the computation broadcast problem presents a fresh opportunity to explore some of the deeper questions in information theory regarding the structure of information, in a setting that is most appealing for its simplicity – involving only 5 random variables: W_1, W_1', W_2, W_2', S.

Appendix: Proofs of Lemma 2 and Lemma 3

Proof of Lemma 2: From the definition of the rank function, there exist $\mu = \text{rank}(\mathbf{A})$ column vectors of the matrix $\mathbf{A}$ that are linearly independent. Denote the matrix formed by these vectors $\mathbf{A}_{sub}$. The column vectors of $\mathbf{A}$ are linear combinations of those of $\mathbf{A}_{sub}$, i.e., $\mathbf{X}^T\mathbf{A}$ are deterministic functions of $\mathbf{X}^T\mathbf{A}_{sub}$. Therefore we have

$$H(\mathbf{X}^T\mathbf{A}) = H(\mathbf{X}^T\mathbf{A}_{sub}) \tag{165}$$

It suffices now to prove that $H(\mathbf{X}^T\mathbf{A}_{sub}) \leq \mu$ and $H(\mathbf{X}^T\mathbf{A}_{sub}) \geq \mu$. It is trivial to see that $H(\mathbf{X}^T\mathbf{A}_{sub}) \leq \mu$ because $\mathbf{X}^T\mathbf{A}_{sub}$ contains only μ elements in $\mathbb{F}_q$ so its entropy cannot be more than μ in q-ary units (uniform distribution maximizes entropy). Next, we show that $H(\mathbf{X}^T\mathbf{A}_{sub}) \geq \mu$. From the definition of the rank function, $\mathbf{A}_{sub}$ contains a square $\mu \times \mu$ invertible sub-matrix. Denote this sub-matrix as $\mathbf{A}_{squ}$. Without loss of generality, assume $\mathbf{A}_{squ}$ is formed by the first μ rows of $\mathbf{A}_{sub}$.

$$\begin{aligned}
H(\mathbf{X}^T\mathbf{A}_{sub}) &\geq H(\mathbf{X}^T\mathbf{A}_{sub} \mid x_{\mu+1}, \cdots, x_{m-1}, x_m) && (166)\\
&= H([x_1, x_2, \cdots, x_\mu]\mathbf{A}_{squ} \mid x_{\mu+1}, \cdots, x_{m-1}, x_m) && (167)\\
&= H(x_1, x_2, \cdots, x_\mu \mid x_{\mu+1}, \cdots, x_{m-1}, x_m) && (168)\\
&= \mu && (169)
\end{aligned}$$

where (168) follows from the fact that $\mathbf{A}_{squ}$ is invertible and applying invertible transformations does not change the entropy, and the last step is due to the condition that $x_1, \cdots, x_m$ are i.i.d. uniform over $\mathbb{F}_q$. This completes the proof of Lemma 2.

Proof of Lemma 3: Lemma 3 follows immediately from Lemma 2. Note that

$$I(\mathbf{X}^T\mathbf{A}; \mathbf{X}^T\mathbf{B}) = H(\mathbf{X}^T\mathbf{A}) + H(\mathbf{X}^T\mathbf{B}) - H(\mathbf{X}^T[\mathbf{A}, \mathbf{B}]) \tag{170}$$

$$= \text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B}) - \text{rank}([\mathbf{A}, \mathbf{B}]) \tag{171}$$

where we have used Lemma 2 in the last step. Therefore $I(\mathbf{X}^T\mathbf{A}; \mathbf{X}^T\mathbf{B}) = 0$ if and only if $\text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B}) = \text{rank}([\mathbf{A}, \mathbf{B}])$, which is in turn equivalent to that $\text{span}(A)$ and $\text{span}(B)$ are independent subspaces. This completes the proof of Lemma 3.

References

[1] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.

[2] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2018.

[3] Q. Yu, N. Raviv, J. So, and A. S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security and privacy," *arXiv preprint arXiv:1806.00939*, 2018.

[4] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning*, 2017, pp. 3368–3376.

[5] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," in *Advances In Neural Information Processing Systems*, 2016, pp. 2100–2108.

[6] H. Sun and S. A. Jafar, "The capacity of private computation," *arXiv preprint arXiv:1710.11098*, 2017.

[7] M. Braverman, "Coding for interactive computation: progress and challenges," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 1914–1921.

[8] T. Lee and A. Shraibman, "Lower bounds in communication complexity," *Foundations and Trends® in Theoretical Computer Science*, vol. 3, no. 4, pp. 263–399, 2009.

[9] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[10] T. Cover, A. E. Gamal, and M. Salehi, "Multiple access channels with arbitrarily correlated sources," *IEEE Transactions on Information theory*, vol. 26, no. 6, pp. 648–657, 1980.

[11] T. Han and M. Costa, "Broadcast channels with arbitrarily correlated sources," *IEEE Transactions on Information Theory*, vol. 33, no. 5, pp. 641–650, 1987.

[12] M. Salehi and E. Kurtas, "Interference channels with correlated sources," in *Proceedings. IEEE International Symposium on Information Theory*. IEEE, 1993, pp. 208–208.

[13] S. S. Pradhan, S. Choi, and K. Ramchandran, "Achievable rates for multiple-access channels with correlated messages," in *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on.* IEEE, p. 108.

[14] E. Tuncel, "Slepian-wolf coding over broadcast channels," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1469–1482, 2006.

[15] D. Gunduz, E. Erkip, A. Goldsmith, and H. V. Poor, "Source and channel coding for correlated sources over multiuser channels," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3927–3944, 2009.

[16] W. Liu and B. Chen, "Interference channels with arbitrarily correlated sources," *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 8027–8037, 2011.

[17] N. Liu, D. Gunduz, A. J. Goldsmith, and H. V. Poor, "Interference channels with correlated receiver side information," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 5984–5998, 2010.

[18] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM'98*, vol. 3, 1998, pp. 1257–1264.

[19] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," in *47th Annual IEEE Symposium on Foundations of Computer Science, 2006. FOCS '06.*, 2006, pp. 197 – 206.

[20] N. Lee, A. G. Dimakis, and R. W. Heath, "Index coding with coded side-information," *IEEE Communications Letters*, vol. 19, no. 3, pp. 319–322, 2015.

[21] S. Miyake and J. Muramatsu, "Index coding over correlated sources," in *Network Coding (NetCod), 2015 International Symposium on.* IEEE, 2015, pp. 36–40.

[22] S. Li, R. Yeung, and N. Cai, "Linear network coding," in *IEEE Trans. on Inform. Theory*, vol. 49, 2003, pp. 371–381.

[23] A. A. Gohari, S. Yang, and S. Jaggi, "Beyond the cut-set bound: Uncertainty computations in network coding with correlated sources," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5708–5722, 2013.

[24] J. Korner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Trans. Inform. Theory*, vol. 25, pp. 219–221, March 1979.

[25] T. Philosof and R. Zamir, "On the loss of single-letter characterization: the dirty multiple access channel," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2442–2454, 2009.

[26] S. S. Pradhan, S. Choi, and K. Ramchandran, "A graph-based framework for transmission of correlated sources over multiple-access channels," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4583–4604, 2007.

[27] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6463–6486, 2011.

[28] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge University Press, 2014.

[29] V. Cadambe and S. Jafar, "Interference Alignment and the Degrees of Freedom of the K user Interference Channel," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[30] S. Jafar, "Interference Alignment: A New Look at Signal Dimensions in a Communication Network," in *Foundations and Trends in Communication and Information Theory*, 2011, pp. 1–136.

[31] Z. Zhang and R. W. Yeung, "On characterization of entropy function via information inequalities," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1440 – 1452, Jul. 1998.